계산과 법연산, 그리고 비밀통신을 강조한

기초정수론

William Stein
강병련 역

August 01, 2019

Contents

서문

이 책은 소수와 합동, 비밀통신, 그리고 타원 곡선에 관한 내용을 아주 치밀하게 엮은 책이다. 이 책은 저자가 하버드, 캘리포니아 대학 샌디에고 캠퍼스, 그리고 워싱톤 대학 등에서 학부 학생들을 가르치는 과정에서 만들어졌다.

체계적인 정수론은, 유클리드가 소수가 무한히 많다는 것과 모든 양의 정수는 소수들의 곱으로 유일하게 인수분해할 수 있다는 산술의 기본 정리를 증명한 때인, 약 300B.C.년 경에 시작되었다. 약 천년이 지나서 (약 972A.D. 경에) 아랍의 수학자들이 어떤 자연수 n이 세 변이 모두 자연수인 직각 삼각형의 면적으로 나타날 수 있는지를 묻는 *합동수 문제*를 생각하였다. 그런 후 또 한 번의 천 년의 세월이 지나 1976년에 Diffi와 Hellman이 공용 채널을 이용하여 두 사람이 사전에 약속한 비밀도 없이 비밀통신을 할 수 있는 공개키 암호(public-key cryptosystem)를 처음으로 소개하였다. 이것과 또 연이어 개발된 여러 공개키 암호들은 디지털 통신의 세계를 혁명적으로 변화시켰다. 1980년대와 1990년대에는 타원곡선이론이, 합동식, 소수 판정, 공개키 암호 생성 및 공격 등의 연구에 새로운 통찰력을 제시하고 또 Andrew Wiles의 Fermat의 마지막 정리의 해결과정에서 중심적인 역할을 함으로써, 정수론에 아주 큰 변화를 가져왔다.

오늘날 순수와 응용 정수론은 동시에 넓고 깊은 이론들의 아주 흥미로운 복합체로서, 새로운 알고리듬과 구체적인 계산으로부터 끊임없이 정보를 얻으면서 발전하고 있다. 합동수 문제가 활발한 연구로 해결 될 가능성이 아주 크며, 소수의 구조에 대한 우리의 이해도 한층 더 깊어졌다. 이 연구들에 대한 우리의 도전 덕에 비밀리 통신하는 우리의 능력도 한층 개선되었다. 이 책의 목표는 독자들을 이런 세상에 좀 더 가깝게 데려가는 것이다.

독자들은 이 책의 모든 연습문제에 꼭 도전해 보기 바란다. 다는 아니지만 일부 문제들의 해답은 이 책의 뒷부분에 있으므로 답도 꼭 확인해 보기 바란

다. 또 이 책의 많은 예들은 공짜로 모두에게 공개되어 있으나 아주 강력한 수학 계산 소프트웨어인 Sage(`http://www.sagemath.org`)를 이용하여 계산을 수행하였다. 독자들은 이 예들을 모두 실행해보고 비슷한 예들로 실험을 반드시 더 해 보길 부탁한다.

사전지식. 독자들은 수학적 증명을 읽고 쓰는 법을 알아야 하고 또 군, 환, 체의 기본 개념을 알아야만 한다. 따라서 이 책은 학부학생들을 위한 책이지만, 보통의 기초정수론의 필요조건보다는 더 많은 것을 필요로 하는 책이다.

표현과 관례. 자연수들의 집합은 $\mathbf{N} = \{1, 2, 3, \ldots\}$으로, 정수들의 집합, 유리수들의 집합, 실수들의 집합, 복소수들의 집합도 $\mathbf{Z}$, $\mathbf{Q}$, $\mathbf{R}$, 그리고 $\mathbf{C}$와 같이 표준 기호를 사용한다. 이 책에서는 기초정리(proposition), 정리(theorem), 보조정리(lemma), 그리고 따름정리(corollary)를 다음과 같이 사용한다. 보통 기초정리는 덜 중요하거나 덜 근본적인 주장들에 사용하고, 정리는 아이디어들의 결합체로서 얻은 주장들에, 보조정리는 뒤에 나올 정리나 기초정리에 이용되는 주장에, 따름정리는 정리와 기초정리, 그리고 보조정리로부터 쉽게 얻어지는 결과들에 사용한다. 좀 어려울 수 있는 문제는 (*)로 표시한다.

감사글 저자는 Brian Conrad, Carl Pomerance, 그리고 Ken Ribet 교수님께 책의 많은 부분을 명확히 해 준 제안들에 감사한다. Baurzhan Bektemirov, Lawrence Cabusora, 그리고 Keith Conrad는 이 책의 원고를 읽어주고 많은 지적을 해 주었고, Carl Witty는 특히 처음 두 단원을 아주 자세하게 지적해 주었다. Frank Calegari은 하버드에서 Math 124 과정을 가르칠 때 이 원고를 사용해 주어 그와 그의 학생들로부터 피드백을 많이 받을 수 있었다. Noam Elkies는 Exercise 4.6를 제안하였다. Seth Kleinerman은 수업 과제로 절 5.4을 작성하였다. Hendrik Lenstra는 그의 인수분해 알고리듬을 구현할 때 도움을 주었다. Michael Abshoff, Sabmit Dasgupta, David Joyner, Arthur Patterson, George Stephanides, Kevin Stern, Eve Thompson, Ting-You Wang, 그리고 Heidi Williams는 여러 잘못을 수정해 주었다. 저자는 또한 Henry Cohn와 David Savitt와의 토론으로부터 많은 도움을 받았다. 저자는 이 책을 준비하기 위하여 Sage[44], emacs, 그리고 LaTeX을 이용하였다.

역자서문

몇 년전 이 책을 본 순간 우리 학생들에게 정수론을 가르칠 때 여기에 소개된 Sage를 이용하는 정수론 강의를 하고 싶다는 생각을 하였다. 유감스럽게도 최근 몇 년간은 정수론을 강의 하지 못하였고, 물론 더 중요한 것은 역자가 직접 Sage를 포함하여 강의를 할 때 어떤 결과가 나타날런지를 검증해 볼 충분한 시간을 가질 수 없었다. 역자의 능력을 생각하여 이제 일의 중요도를 바꾸니 새로운 강의를 시도해 볼 자신이 생겼다.

그 첫 번째 시도로 이 책을 번역하여 정수론과 암호론, 대수학 특강을 강의할 때 참고 도서로 삼으려고 한다. 이 번역이 나와 우리 학생들에게 도움이 되길 바란다.

이 책의 번역은 다음과 같이 시작되었다. 역자가 현재 집필 예정인 기초정수론 책에 이 책의 Sage코드를 이용할 수 있는지에 대한 저자의 뜻을 알기 위해 연락을 하였고 저자는 흔쾌히 허락하였다. 독립적으로 이 책이 한국어로 번역되는 것을 원한다면 역자가 할 수 있다고 했고, 저자도 아주 좋다고 하였다.

이에 나의 책에 이 책의 Sage코드를 소개하는 것보다 먼저 저자의 저서의 내용을 그의 이름으로 먼저 소개하는 것이 옳다고 생각하던 차에 충남대학교 2016년 강의교재개발을 계기로 번역을 시작하였다.

아마 역자가 실제 강의를 할 때는 다시 우리 학생들에 맞는 내용으로 편역되어 있지 않을까 한다.

이 책은 다른 기초 정수론 책보다는 속도감 있게 진행된다. 대수학에서 다루는 군, 환, 체 등의 용어들을 소개는 하지만, 증명에서 대수학의 성질을 사용하는 경우는 극히 드물다(3단원까지는 전혀 없음). 학부 정수론 교재로, 혹은 암호론, 대수학의 계산 가능한 영역에서 좋은 참고도서가 되리라 믿고 번역하였다.

번역서가 도움이 되는 모든 학생들이 좀 더 편하게 부담없이 참고할 수 있도록 책으로 출간하기 보다는 저자의 홈페이지에 번역본을 올리게 되었다. 일부 학생들에게라도 이 번역서가 도움이 된다면 역자는 굉장히 행복할 것 같다. 비록 저자의 홈페이지에 온라인으로 올리긴 했지만, 연습문제의 번역을 이 책을 읽는 각 학생들의 몫으로 남긴 것을 제외하고는, 언제라도 출판 가능한 완성된 번역임을 밝힌다.

한글 번역본에 대한 의견은 plkang@cnu.ac.kr로 보내주기 바란다.

연구년 덕분에 이 책의 번역에 수 개월의 시간을 집중할 수 있었다. 충남대에 감사한다.

2017년 1월 12일

마지막 수정: 2019년 7월 22일

1
소수
(Prime Numbers)

$100 = 2^2 \cdot 5^2$으로 쓸 수 있는 것처럼 모든 양의 정수는 소수들의 곱셈으로 유일하게 표현할 수 있다. 익숙한 이 사실의 증명은 그렇게 쉽지만은 않다. 심지어 더 놀라운 사실은 어떤 1000자리 수의 경우에는 소수들의 곱으로 표현하는 방법을 찾는 것조차도 현재의 기술로는 불가능해보이는데, 그런데 이 기술이 인터넷에서 물건을 사는 수백만의 사람들에 의해서 매일 사용되고 있다는 것이다.

소수들로 모든 정수들을 표현할 수 있으므로 소수들이 얼마나 많이 또 어떻게 정수들 사이에 분포하는지 궁금해지는 것은 아주 자연스러운 호기심이 아닐까?

> "There are two facts about the distribution of prime numbers. The first is that, [they are] the most arbitrary and ornery objects studied by mathematicians: they grow like weeds among the natural numbers, seeming to obey no other law than that of chance, and nobody can predict where the next one will sprout. The second fact is even more astonishing, for it states just the opposite: that the prime numbers exhibit stunning regularity, that there are laws governing their behavior, and that they obey these laws with almost military precision."
>
> — Don Zagier[55]

정수론에서 가장 유명한 미해결 문제인 리만 가설(Riemann hypothesis)은 소수의 분포에 대한 정확한 답을 주고 있다고 많은 수학자들이 생각하는 가설이다.

이 단원에서는 소수, 소인수 분해, 그리고 소수의 분포와 같은 주제들을 잘 엮어 정수론을 연구하는 기초를 소개한다. 1.1절에서는 2이상의 모든 정수는 소수들이 곱셈으로 표현할 수 있다는 소인수분해를 엄밀하게 증명하고, 소수들이 곱으로 표현하면 상금을 받을 수 있는 특별한 정수들을 몇 가지 소개한다. 1.2절에서는 소수가 무한히 많다는 유클리드(Euclid)의 증명으로 시작해서 소수들의 집합에 대하여 논하고, 또 알려진 가장 큰 소수에 관한 이야기를 소개한다. 마지막으로 소수 정리와 리만 가설을 통해 소수의 분표에 대해 알아본다.

1.1 소인수분해

1.1.1 소수

*자연수*들의 집합과 *정수*들의 집합은 각각 $\mathbf{N}$, $\mathbf{Z}$로 표현한다.

$$\mathbf{N} = \{1, 2, 3, 4, \ldots\},$$

$$\mathbf{Z} = \{\ldots, -2, -1, 0, 1, 2, \ldots\}.$$

정의 1.1.1 (나눈다)**.** a와 b가 정수일 때 $b = ac$인 적당한 정수 c가 존재하면 a는 b를 **나눈다**고 하고 $a \mid b$로 쓴다. 이 경우 a는 b의 **약수(divisor)**라고 한다. $b = ac$인 정수 c가 존재하지 않으면 a는 b를 **나누지 않는다**고 하고 $a \nmid b$로 쓴다.

예를 들어 $2 \mid 6$이고 $-3 \mid 15$이다. 또 모든 정수는 0을 나누고 0은 오직 0만 나눈다. 그러나 3은 7을 정수에서는 나누지 않는다.

참조 1.1.2. 러시아의 정수론 책에서는 a가 b를 나눈다를 보통 $b \vdots a$ 로 표현한다.

정의 1.1.3 (소수와 합성수)**.** n은 1보다 큰 정수이다. 이 때 n의 양의 약수가 1과 n뿐이면 **소수(prime)**, 그렇지 않으면 **합성수**라고 부른다.

1은 소수도 합성수도 아니다. 차례대로 소수들을 나열하면

$$2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, \ldots,$$

이고 차례대로 합성수를 나열하면 다음과 같다.

$$4, 6, 8, 9, 10, 12, 14, 15, 16, 18, 20, 21, 22, 24, 25, 26, 27, 28, 30, 32, 33, 34, \ldots.$$

참조 1.1.4. J. H. Conway는 [11, viii]에서 -1은 소수로 봐야하고, Lehmer는 1214년 소수표 [30]에서 1를 소수로 간주하였다. 이 책에서는 1은 소수도 합성수도 아닌 수로 본다.

SAGE 예 1.1.5. Sage를 이용하여 a와 $b-1$사이의 모든 소수를 찾을 수 있다.

```
sage: prime_range(10,50)
[11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47]
```

a와 $b-1$사이의 모든 합성수도 찾을 수 있다.

```
sage: [n for n in range(10,30) if not is_prime(n)]
[10, 12, 14, 15, 16, 18, 20, 21, 22, 24, 25, 26, 27, 28]
```

모든 자연수는 유일한 방법으로 소수로부터 만들어진다:

정리 1.1.6 (산술의 기본 정리 (Fundamental Theorem of Arithmetic))**.** *1보다 큰 모든 자연수는 소수들의 곱으로 표현할 수 있고 그 방법은 차례를 무시하면 유일하다.*

이 때 소수는 한 개의 소수의 곱이다. 1은 공집합의 원소들의 곱으로 약속할 수 있다.

참조 1.1.7. 정리 1.1.6는 1.1.4절에서 증명하고자 하는데 여러분이 생각하는 것보다는 까다롭다. 일반적으로 인수분해가 되면 그 방법은 한 가지 뿐일 것이라고 생각하기가 쉬우나 반드시 그렇지 않을 수도 있다. 집합

$$\mathbf{Z}[\sqrt{-5}] = \{a + b\sqrt{-5} : a, b \in \mathbf{Z}\} \subset \mathbf{C},$$

에서 6은 두 가지 방법으로 인수분해된다. ($\mathbf{C}$는 복소수들의 집합이다.)

$$6 = 2 \cdot 3 = (1 + \sqrt{-5}) \cdot (1 - \sqrt{-5}).$$

1.1.2 최대공약수

최대공약수의 개념을 이용하여 소수에는 다음 성질

$$p \mid ab \;\Rightarrow\; p \mid a \text{ 혹은 } p \mid b$$

이 성립함을 보이자. 이는 정리 1.1.6의 증명에서 핵심적인 역할을 한다.

정의 1.1.8 (최대공약수)**.** 둘 다 0이 아닌 두 정수 a, b의 모든 공통 약수 중에서 가장 큰 수를 a와 b의 **최대공약수 (Greatest Common Divisor)**라고 하고 $\gcd(a, b)$로 쓴다. 즉,

$$\gcd(a, b) = \max\{d \in \mathbf{Z} : d \mid a \text{ and } d \mid b\}.$$

$\gcd(0, 0) = 0$으로 약속한다.

예를 들어 $\gcd(1, 2) = 1$, $\gcd(6, 27) = 3$ 이며, 모든 정수 a에 대하여 $\gcd(0, a) = \gcd(a, 0) = a$이 성립한다.

$a \neq 0$이면 $\gcd(a, b)$는 존재한다. 왜냐하면 $d \mid a$이면 $d \leq |a|$이고 $|a|$보다 작거나 같은 자연수는 $|a|$ 개뿐이기 때문이다. 같은 이유로 $b \neq 0$인 경우도 gcd는 존재한다.

보조정리 1.1.9. *임의의 정수 a와 b에 대하여 다음이 성립한다.*

$$\gcd(a, b) = \gcd(b, a) = \gcd(\pm a, \pm b) = \gcd(a, b - a) = \gcd(a, b + a).$$

증명 다른 경우들도 같은 방법으로 증명할 수 있으므로 $\gcd(a,b)=\gcd(a,b-a)$만 보이자. d가 a,b의 공약수라면, 즉 $d\mid a$, $d\mid b$라면 $dc_1=a$과 $dc_2=b$를 만족하는 정수 c_1과 c_2가 존재한다. 그러면 $b-a=dc_2-dc_1=d(c_2-c_1)$이므로 $d\mid b-a$이다. 따라서 $\gcd(a,b)\le\gcd(a,b-a)$. 왜냐하면, a,b의 공통약수들이 a와 $b-a$의 공약수가 되기 때문이다. a를 $-a$로 b를 $b-a$로 바꾸면 $\gcd(-a,b-a)\le\gcd(-a,b)$이므로 $\gcd(a,b-a)=\gcd(-a,b-a)\le\gcd(-a,b)=\gcd(a,b)$가 되어 $\gcd(a,b)=\gcd(a,b-a)$가 성립한다. □

보조정리 1.1.10. *$a,b,n\in\mathbf{Z}$일 때 $\gcd(a,b)=\gcd(a,b-an)$이 성립한다.*

증명 보조정리 1.1.9를 계속하여 사용하면

$$\gcd(a,b)=\gcd(a,b-a)=\gcd(a,b-2a)=\cdots=\gcd(a,b-an).$$

□

정리 1.1.6를 증명했다고 잠시 가정하자. 그러면 $\gcd(a,b)$를 계산하기 위한 소박한 방법은 정리 1.1.6를 이용하여 a와 b를 소수들의 곱으로 표현하고 그것들로부터 $\gcd(a,b)$를 소수들의 곱으로 표현하는 것이다. 예를 들어, $a=2261, b=1275$이면 $a=7\cdot\mathbf{17}\cdot 19, b=3\cdot 5^2\cdot\mathbf{17}$이다. 따라서 $\gcd(a,b)=17$이다. 그러나 a와 b를 소인수분해하지 않고도 아래의 알고리듬 1.1.13을이용하여 $\gcd(a,b)$를 구할 수 있다.

알고리듬 1.1.13의 동기 부여를 위하여 $\gcd(2261,1275)$를 위에서 보인 것과 다른 방법으로 계산한다. 먼저 다음 유용한 사실을 기억해내자.

기초정리 1.1.11. *a,b는 정수이고 $b\neq 0$이다. 그러면 $a=bq+r$이 되는 정수 q와 $0\le r<|b|$인 정수 r이 존재하고 또 유일하다.*

증명 존재성은 a와 b가 양수인 경우만 증명한다. (일반적인 경우는 독자의 몫으로 남긴다.) Q는 $a-bn\ge 0$인 음수가 아닌 정수 n들의 집합이다. 그러면 $0\in Q$이므로 Q는 공집합이 아니다. 또 $a-bn\ge 0$은 $n\le a/b$와 동치이므로 Q의 모든 원소는 a/b보다 작거나 같다. 그러므로 Q에서 가장 큰 정수 q를 선택할 수 있다. 이제 $r=a-bq$라면 $r<b$이다. 왜냐하면 $r=a-bq\ge b$이면 $a-b(q+1)\ge 0$이 되어 q보다 큰 $q+1$이 Q에 속하여 모순이다. 따라서 우리가 원하는 조건을 만족하는 r,q가 존재한다.

유일성: $a=bq+r=bq_1+r_1$이고 $0\le r<|b|$, $0\le r_1<|b|$이라면

$$b(q-q_1)=r_1-r \tag{1.1.1}$$

이다. 만약 $q\neq q_1$이라면 $|b(q-q_1)|\ge|b|$인데 $-|b|<r_1-r_2<|b|$이므로 $b(q-q_1)$과 r_1-r이 같을 수 없다. 따라서 $q=q_1$이다. 이를 식 1.1.1에 대입하면 $r_1-r=0$이다. 따라서 위의 조건을 만족하는 q와 r은 유일하다.

□

이 책에서는 **알고리듬(algorithm)**은 컴퓨터 프로그램에서 입력 작업이 유효하면 반드시 결과가 나오는 특정 작업을 수행케하는 유한 개의 지시의 나열이다. 가끔은 알고리듬은 여기서 정의한 것 보다는 좀 더 느슨한 (혹은 좀 더 정확한) 의미로 쓰이기도 하지만 이 책에서는 이 정의로 충분하다.

알고리듬 1.1.12 (나눗셈 알고리듬 (division algorithm))**.** a, b는 정수이고 $b \neq 0$이다. 그러면 이 알고리듬은 $a = bq + r$이 되는 q와 $0 \leq r < |b|$를 만족하는 r을 계산한다.

알고리듬 1.1.12은 우리가 잘 아는 세로로 길게 하는 긴 나눗셈이므로 각 단계를 서술하지는 않을 것이다. 그러나 어떤 경우에든 a가 음수인 경우에도 이 알고리듬은 나머지를 항상 양수로 만든다는 점에서는 우리가 익숙한 나눗셈과는 조금 다를 수 있다.

나눗셈 알고리듬을 계속 시용하여 $\gcd(2261, 1275)$를 계산하자. 2261를 1275로 나누면

$$2261 = 1 \cdot 1275 + 986,$$

이므로 $q = 1$이고 $r = 986$이다. 자연수 d가 2261과 1275을 다 나누면 , d는 이 두 수의 차인 986을 나누면서 여전히 d는 1275도 나눈다. 반면에 d기 1275와 986을 나누면, d는 두 수의 합도 나눈다. 따라서 다음 과정이 성립한다.

$$\gcd(2261, 1275) = \gcd(1275, 986).$$

이 등식은 보조정리 1.1.9로부터도 얻을 수 있다. 이 과정을 계속하면

$$1275 = 1 \cdot 986 + 289,$$

이므로 $\gcd(1275, 986) = \gcd(986, 289)$. 계속하면

$$\begin{aligned} 986 &= 3 \cdot 289 + 119 \\ 289 &= 2 \cdot 119 + 51 \\ 119 &= 2 \cdot 51 + 17. \end{aligned}$$

을 얻으므로 $\gcd(2261, 1275) = \cdots = \gcd(51, 17)$이 되는데, $17 \mid 51$이므로 $\gcd(51, 17)$은 17이다. 따라서

$$\gcd(2261, 1275) = 17.$$

지겨운 산술 계산이긴 하지만 이 계산은 상당히 조직적이며 또 (아주 큰 수의 경우는 소수의 곱으로 표현하는 빠른 방법을 알지도 못하지만) 주어진 두 수를 소수의 곱으로 표현할 필요도 없었다.

알고리듬 1.1.13 (최대공약수 계산)**.** 정수 a, b에 대하여 이 알고리듬은 $\gcd(a, b)$를 계산한다.

1. [$a > b > 0$로 가정] $\gcd(a, b) = \gcd(|a|, |b|) = \gcd(|b|, |a|)$이므로 a와 b를 a의 절대값과 b의 절대값으로 각각 바꾼다. $a = b$이면 a를 출력(output)하고 프로그램은 끝난다. 필요하다면 a와 b를 교환하여 $a > b$로 가정한다. $b = 0$이면 a를 출력한다.
2. [몫과 나머지] 알고리듬 1.1.12을 사용하여 $a = bq + r$로 쓴다. 이 때 $0 \leq r < b$이고 $q \in \mathbf{Z}$이다.
3. [끝?] $r = 0$이면 $b \mid a$이므로 b를 출력하고 끝난다.

4. [바꾸기와 되풀이] $a \leftarrow b$, $b \leftarrow r$로 바꾼 후 단계 2로 간다.

증명 보조정리 1.1.9–1.1.10 로부터 $\gcd(a,b) = \gcd(b,r)$가 성립하므로 gcd는 단계 4에서 바뀌지 않는다. 또 나머지는 계속 감소하는 음수가 아닌 정수이므로 이 알고리듬은 유한 단계에서 끝난다. □

예 1.1.14. $a = 15$, $b = 6$로 놓자.

$$\begin{aligned} 15 &= 6 \cdot 2 + 3 \qquad \gcd(15,6) = \gcd(6,3) \\ 6 &= 3 \cdot 2 + 0 \qquad \gcd(6,3) = \gcd(3,0) = 3 \end{aligned}$$

10배 큰 수들이라 해도 이 계산법은 마찬가지로 쉬운데, 이 관찰은 아래 정리 1.1.19의 증명에서 중요한 역할을 한다.

예 1.1.15. $a = 150, \quad b = 60$으로 놓자

$$\begin{aligned} 150 &= 60 \cdot 2 + 30 \qquad \gcd(150,60) = \gcd(60,30) \\ 60 &= 30 \cdot 2 + 0 \qquad \gcd(60,30) = \gcd(30,0) = 30 \end{aligned}$$

SAGE 예 1.1.16. Sage에서는 `gcd`가 최대공약수를 구하는 명령어이다. 예를 보자.

```
sage: gcd(97,100)
1
sage: gcd(97 * 10^15, 19^20 * 97^2)
97
```

보조정리 1.1.17. *a, b, n가 정수일 때*

$$\gcd(an, bn) = \gcd(a,b) \cdot |n|$$

이 성립한다.

증명 예 1.14와 1.1.15로 부터 짐작할 수 있듯이 유클리드의 알고리듬으로 gcd를 계산할 때 $\gcd(an,bn)$의 각 단계의 식은 $\gcd(a,b)$의 해당 단계의 식에 n을 곱한 식이다. 간단히 증명하기 위하여 a와 b 모두 양수라고 가정하고 $a+b$에 관한 수학적 귀납법을 사용한다. $a+b = 2$이면 $a = b = 1$이므로 위의 명제는 성립한다. a, b는 $a \geq b$인 임의의 자연수라고 하자. $a = bq + r,\ 0 \leq r < b$인 정수 q와 r이라면 보조정리들 1.1.9–1.1.10에 의하여, $\gcd(a,b) = \gcd(b,r)$이다. 식 $a = bq + r$에 n을 곱하면 $an = bnq + rn$이므로, $\gcd(an,bn) = \gcd(bn,rn)$. 또

$$b + r = b + (a - bq) = a - b(q-1) \leq a < a + b,$$

이므로 수학적 귀납법에 의하여 $\gcd(bn,rn) = \gcd(b,r) \cdot |n|$이다. $\gcd(b,r) = \gcd(a,b)$이므로, 이 보조정리는 성립한다. □

보조정리 1.1.18. *두 정수 a, b의 공약수는 $\gcd(a,b)$의 약수이다. 즉, 정수 n이 $n \mid a$, $n \mid b$이면 $n \mid \gcd(a,b)$이다.*

증명 $n \mid a$, $n \mid b$이므로 $a = nc_1$ $b = nc_2$인 정수 c_1과 c_2가 존재한다. 보조정리 1.1.17로부터 $\gcd(a,b) = \gcd(nc_1, nc_2) = n\gcd(c_1, c_2)$이므로 n은 $\gcd(a,b)$를 나눈다. □

알고리듬 1.1.13으로 어떤 소수가 두 정수의 곱을 나누면 그 소수는 두 수 중의 하나는 반드시 나누어야하는 사실을 증명할 수 있다. 이는 소인수 분해가 유일함을 증명하는데 핵심적인 역할을 한다.

정리 1.1.19 (Euclid)**.** *p는 소수이고 $a, b \in \mathbf{N}$이다. 만약 $p \mid ab$이면 $p \mid a$ 혹은 $p \mid b$이다.*

산술의 기본정리에 이미 여러분들이 익숙하기 때문에 정리 1.1.19가 직관적으로 자명하다고 혹 생각할수도 있지만, 정리 1.1.19는 산술의 기본정리 (정리 1.1.6)를 증명할 때 꼭 필요하다.

증명 [정리 1.1.19의 증명] 이미 $p \mid a$이면 증명할 필요가 없다. 1하고 p만이 p를 나누므로 $p \nmid a$이면 $\gcd(p,a) = 1$이다. 보조정리 1.1.17로부터, $\gcd(pb, ab) = b\gcd(p,a) = b$인데 $p \mid pb$ 이고 또 가정에 의하여 $p \mid ab$이므로, (보조정리 1.1.17)를 사용하여

$$p \mid \gcd(pb, ab) = b\gcd(p,a) = b \cdot 1 = b$$

가 성립한다. □

1.1.3 정수들은 소수들의 곱으로 인수분해된다.

이 절에서는 모든 자연수는 소수들의 곱으로 표현할 수 있음을 보인다. 그 후 실제 소수들의 곱으로 표현하는 것은 얼마나 어려운 작업인지를 논의한다. 인수분해의 유일성은 1.1.4절에서 증명한다.

첫 번째 예는 $n = 1275$의 소인수분해이다. n의 각 자릿수의 합이 3의 배수이므로 n은 3의 배수이다. (기초정리 2.1.9 참조). $n = 3 \cdot 425$가 되는데 425 5의 배수이므로 (왜냐하면 일의 자릿수가 5의 배수이므로) $1275 = 3 \cdot 5 \cdot 85$가 된다. 다시 85는 5의 배수이므로, $1275 = 3 \cdot 5^2 \cdot 17$임을 알 수 있다. 3, 5, 17이 모두 소수이므로 $3 \cdot 5^2 \cdot 17$은 1275의 소인수 분해이다. 이 과정을 일반화하면 다음 명제를 증명할 수 있다.

기초정리 1.1.20. *모든 자연수는 소수의 곱이다.*

증명 자연수 n에 대한 귀납법을 사용한다. $n = 1$이면 n은 소수의 공집합의 곱이다. n이 소수이면, 한 개의 소수의 곱이다. n이 합성수이면 n은 더 작은 자연수 a, b의 곱이다. 귀납적 가정에 의하여 a 와 b가 소수의 곱이므로 n도 소수의 곱이다. □

이제 모든 자연수는 소수의 곱임을 알 때 다음 두 질문은 아주 자연스럽다. (1) 소수의 곱으로 표현하는 방법은 하나일까? (2) 얼마나 빠르게 자연수를 소수의 곱으로 표현할 수 있을까? (1)은 예를 들어 1275를 다른 방법으로 인수분해를 시도했을 때 어떻게 되는지를 확인해 보면 된다. 시도해보면:

$1275 = 5{\cdot}255$이고, $255 = 5{\cdot}51$이고 $51 = 17{\cdot}3$이므로, 정리 1.1.6에서 주장하였듯이 결국 같은 결과를 얻는다. 자연수의 소인수분해의 유일성은 Section 1.1.4에서 증명한다.

SAGE 예 1.1.21. Sage에서 `factor`는 주어진 자연수를 소인수분해하는 명령어이다. 계산 창에 소수의 거듭제곱의 곱으로 나타낸다. 예를 보자.

```
sage: factor(1275)
3 * 5^2 * 17
sage: factor(2007)
3^2 * 223
sage: factor(31415926535898)
2 * 3 * 53 * 73 * 2531 * 534697
```

질문 (2)와 관련해서는 정수의 소인수분해 알고리듬들이 존재한다. 얼마나 빠른 소인수분해 알고리듬이 존재할 수 있는지는 아직 해결하지 못한 중요한 문제이다. n을 인수분해하는 알고리듬이 **다항식 시간 (polynomial time)**이라는 것은 다항식 $f(x)$가 존재하여 이 알고리듬이 자연수 n을 인수분해하는데 필요한 단계의 수가 $f(\log_{10}(n))$보다 작을 때를 의미한다. $((\log_{10}(10^k) = k))$에서 짐작할 수 있듯이 $\log_{10}(n)$은 n의 자릿수의 근삿값이다.

Open Problem 1.1.22. *정수 n의 다항식 시간 인수분해 알고리듬이 존재하는가?*

Peter Shor는 [46]에서 양자 컴퓨터에서의 정수의 다항식 시간 인수분해 알고리듬을 고안하였다. 2001년 IBM 연구원들이 Shor의 알고리듬을 구현하여 15를 인수분해하긴 했었지만 ([34, 24]) 이 사실 외에는 이 책에서는 이 알고리듬을 더 이상 논하지는 않을것이다. 훨씬 큰 양자 컴퓨터를 만드는 일은 굉장히 어려운 작업으로 보인다.

어떤 큰 수를 인수분해 하면 돈을 벌 수도 있다. 어떤 큰 정수의 인수분해가 쉽다면 많은 암호체계(cryptosystems)가 쉽게 깨질 수 있다. 그래서, 아무도 인수분해가 어렵다는 것을 증명하지는 않았으므로 인수분해가 어렵다는 사실에 대한 확신을 높이기 위하여 어떤 정수의 인수분해에 상금을 걸기도 한다. 다음은 최근 인수분해하는데 $10,000의 상금이 걸렸던 자릿수가 174인 정수이다([42] 참조).

$$188198812920607963838697239461650439807163563379417382700763356422988859715234665485319060606504743045317388011303396716199692321205734031879550656996221305168759307650257059$$

이 수는 2진법으로 자릿수가 576이라 RSA-576으로 알려진 수이다. (이진법 수에 대한 더 많은 정보는 절 2.3.2을 참조) 이 수는 2003년 12월 독일연방 정보기술보안부(German Federal Agency for Information Technology Security)

에서 인수분해하였다([53] 참조)

$$3980750864240649373971255005503864911990643623425267084063851895759463889571261768583317 \times 4727721461074353025362230719730482246329146953020971164598521711305207112563636590397527$$

그 전의 RSA 도전 수는 155-자릿수였다.

$$1094173864157052742180970732204035761200373294544920599091384213147634998428893478471799725789126733249762575289978183379707653724402714674353159335433389 7.$$

이 수는 1999년 8월 22일에 인수분해되었는데 16명의 연구그룹이 292 개의 컴퓨터를 연결하여 얻은 결과였다. ([1] 참조) 그들이 얻은 결과는 RSA-155는 다음 78-자리의 두 소수의 곱이라는 것이다:

$$p = 102639592829741105772054196573991675900716567808038066803419335217907113077 79$$

$$q = 106603488380168454820927220360012878679207958575989291522270608237193062808643.$$

다음 RSA도전은 F. Bahr, M. Boehm, J. Franke 와 T. Kleinjun이 2005년 11월 인수분해할 때까지 $20,000불의 상금이 걸려 있었던 RSA-640이었다. 다음은 RSA-640과 이 수의 인수 중 하나이다.

$$3107418240490043721350750035888567930037346022842727545720161948823206440518081504556346829671723286782437916272838033415471073108501919548529007337724822783525742386454014691736602477652346609,$$

$$1634733645809253848443133883865090859841783670033092312181110852389333100104508151212118167511579.$$

(이 팀은 663 비트 RSA 도전수도 인수분해하였다.)

현재$30,000불을 벌기 위해 도전할 수 있는 가장 작은 수는 RSA-704이다.

$$7403756347956171282804679609742957314259318888923128908493623263897276503402826627689199641962511784399589433050212758537011896809828673317327310893090055250511687706329907239638078671008609696253793465056379635 9$$

SAGE 예 1.1.23. Sage를 사용하면 위의 수 RSA-704는 212자릿수이며 합성수임을 확인할 수 있다.

```
sage: n = 74037563479561712828046796097429573142593188888\
...92312890849362326389727650340282662768919964196251117\
...84399589433050212758537011896809828673317327310893 09\
...00552505116877063299072396380786710086096962537934 65\
...0563796359
sage: len(n.str(2))
704
sage: len(n.str(10))
212
sage: n.is_prime()              # this is instant
False
```

이 RSA수들은 number field sieve로 불리는 알고리듬을 이용하여 인수분해되었는데([33]), 이 알고리듬은 일반적인 목적의 소인수분해에 가장 효율적인 알고리듬으로 알려져 있다. number field sieve 알고리듬이 어떻게 작동하는지를 성명하는 것은 이 책의 범위를 넘어선다. 그러나 number field sieve 알고리듬은 타원 곡선을 아주 많이 이용하는 인수분해 알고리듬이므로 절 6.3에서 설명할 것이다.

1.1.4 산술의 기본정리

우리는 이제 다음 아이디어들을 이용하여 정리 1.1.6를 증명할 준비가 다 되었다. 정수 n을 두 가지 방법으로 인수분해하였다고 가정하자. 정리 1.1.19를 이용하여 두 인수분해에서 공통인 소수들을 하나씩 소거한다. 그러면 결국 각각의 인수분해는 정확히 같은 소수들로 구성되어 있음을 발견하는 것이다. 아래에 자세히 설명한다.

증명 $n = 1$이면 소수의 공집합의 곱이므로 유일하다. 따라서 $n > 1$이라고 가정한다. 기초정리 1.1.20에 의하여

$$n = p_1 p_2 \cdots p_d$$

인 소수 $p_1, \ldots, p_d$ 이 존재한다. 만약

$$n = q_1 q_2 \cdots q_m$$

이 되는 또 다른 소수들 $q_1, \ldots, q_m$이 존재한다고 하자. 그러면

$$p_1 \mid n = q_1(q_2 \cdots q_m)$$

이므로 Euclid의 정리에 의해 $p_1 = q_1$이거나 $p_1 \mid q_2 \cdots q_m$인데 이를 계속하면 p_1는 적당한 q_i와 같아진다. 이제 p_1와 q_i를 소거하고 위에서 한 일들을 p_2, 그리고 $p_3, \ldots$등을 가지고 반복하면 차례를 바꾸는 것 외에는 두 가지 인수분해가 같음을 알아낸다. □

1.2 소수들의 열

이 절에서는 다음 세 가지 질문에 대한 답을 찾고자 한다:

1. 소수는 무한히 많은가?
2. 정수 a, b를 임의로 선택했을 때 $ax+b$ 꼴의 소수는 얼마나 많이 존재할까?
3. 자연수를 차례로 배열할 때 소수들은 어떻게 분포할까?

먼저 소수는 무한히 많음을 보이고, $\gcd(a,b) = 1$이면 $ax+b$가 소수인 x가 무한히 많이 존재한다는 Dirichlet의 정리를 소개한다. 그리고 x보다 작은 소수는 근사적으로 $x/\log(x)$ 개 정도 존재한다는 소수 정리(Prime Number Theorem), 마지막으로 이 소수의 개수에 관한 공식과 리만 가설(Riemann Hypothesis)과의 관계에 대해 논한다.

1.2.1 소수는 무한히 많다

아래 표의 왼쪽은 소수이고 오른쪽의 식은 서로 유사한 형태로 만들어진 식이다. 물론 이 현상은 무한히 계속되지는 않지만 유사한 결과를 준다.

$$\begin{aligned} 3 &= 2+1 \\ 7 &= 2\cdot 3+1 \\ 31 &= 2\cdot 3\cdot 5+1 \\ 211 &= 2\cdot 3\cdot 5\cdot 7+1 \\ 2311 &= 2\cdot 3\cdot 5\cdot 7\cdot 11+1 \end{aligned}$$

정리 1.2.1 (Euclid)**.** *소수는 무한히 많다.*

증명 $p_1, p_2, \ldots, p_n$이 소수라고 하자. 그러면 다음 방법으로 항상 새로운 소수 p_{n+1}를 만들 수 있다.

$$N = p_1 p_2 p_3 \cdots p_n + 1 \tag{1.2.1}$$

을 정의하자. 기초정리 1.1.20에 의해 $N > 1$이므로 한 개 이상의 소수의 곱으로 표현하면

$$N = q_1 q_2 \cdots q_m.$$

이 때 q_i는 소수이고 $m \geq 1$이다. 만약 $q_1 = p_i$이면, $p_i \mid N$이다. 그런데 (1.2.1)으로부터 $p_i \mid N-1$이므로 $p_i \mid 1 = N-(N-1)$을 얻는데 이는 모순이다. 따라서 p_i는 N의 소인수분해에 나타날 수 없으므로 $p_{n+1} = q_1$는 우리가 처음 시작한 소수표인 $p_1, \ldots, p_n$에 없던 새로운 소수이다. 유한개의 소수를 주면 항상 새로운 소수를 얻을 수 있으므로 소수는 무한하다. □

처음 6 개의 소수 2, 3, 5, 7, 11, 13을 곱한 후 1을 더한 수

$$2\cdot 3\cdot 5\cdot 7\cdot 11\cdot 13+1 = 30031 = 59\cdot 509$$

는 그 자체는 소수가 아니지만 새로운 소수에 의해 나누어진다.

Joke 1.2.2 (Hendrik Lenstra)**.** *합성수는 무한히 많다. 증명.* 새 합성수를 만들기 위하여 처음 n개의 합성수를 곱한 후 1을 더하지 않으면 된다.

1.2.2 소수 세기

이 절에서는 정수 n까지의 모든 소수를 계산하기 위하여 (합성수를) 체로 쳐내는 과정을 소개한다. 먼저 1을 제외한 n까지의 모든 수를 적은 후 2가 소수임을 확인 후 2의 배수를 모두 걸러낸다. 이제 남은 수의 첫 번째 수인 3은 소수이므로 3 다음의 모든 3의 배수를 지운다. 이 과정을 반복하면 n 까지의 모든 소수들을 얻는다. 다음은 이 과정의 공식 알고리듬이다.

알고리듬 1.2.3 (소수체 (Prime Sieve))**.** 이 알고리듬은 자연수 n까지의 모든 소수들을 계산한다.

1. [Initialize] $X = [3, 5, \ldots]$는 3과 n사이의 모든 홀수들이고, $P = [2]$는 지금까지 발견한 소수들이다.
2. [Finished?] p는 X의 첫 번째 원소이다. 만약 $p > \sqrt{n}$이면, X의 모든 원소를 P에 넣고 끝난다. 그렇지 않으면 p를 P에 첨가한다.
3. [Cross Off] X는 X에서 p의 배수들을 뺀 집합으로 바꾼 후 단계 2로 간다.

예를 들어 40까지의 소수들을 이 체를 이용하여 구해보자. $P = [2]$,

$$X = [3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25, 27, 29, 31, 33, 35, 37, 39]$$

로 시작한다. 3을 P에 첨가하고 3의 배수들을 X에서 지운 후 새로운 X를 얻는다.

$$X = [5, 7, 11, 13, 17, 19, 23, 25, 29, 31, 35, 37].$$

5을 P에 첨가하고 5의 배수들을 지운 후 새 X는 $X = [7, 11, 13, 17, 19, 23, 29, 31, 37]$. 그런데 $7^2 > 40$이므로 X를 모두 P에 첨가하면

$$P = [2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37]$$

가 된다. 따라서 40이하의 모든 소수는

$$2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37.$$

이다.

증명 [알고리듬 1.2.3의 증명] 이 알고리듬에서 분명하지 않은 것은, 단계 2에서, X의 첫 원소 a가 $a > \sqrt{n}$를 만족하면 X의 모든 원소가 소수라는 부분이다. 이것을 증명하기 위해, $m \in X$라고 가정하자. 그러면 $\sqrt{n} < m \leq n$이고 m은 $\sqrt{n}$보다 작거나 같은 어떤 소수로도 나누어지지 않는다. 이제 $m = \prod p_i^{e_i}$으로 쓸 수 있고, 이 때 p_i는 소수이고 $p_1 < p_2 < \ldots$으로 가정할 수 있다. 그러면 p_1을 포함하여 모든 $p_i > \sqrt{n}$이다. 따라서 만약 m의 인수분해에 두 개 이상의 소수가 필요하다면, 즉 소수가 아니라면, $m > \sqrt{n} \cdot \sqrt{n} = n$이므로 모순이다. □

1.2.3 알려진 가장 큰 소수

비록 정리 1.2.1가 소수는 무한히 많이 존재한다는 사실을 말해주기는 하지만, '알려진 가장 큰 소수가 얼마지?'는 여전히 의미 있는 질문이다.

메르센 소수(Mersenne prime)는 $2^q - 1$ 형태의 소수이다. [6]에 따르면 2007년 3월 기준으로 가장 큰 소수는 자릿수가 9,808,358인 44 번째 메르센 소수로 알려진

$$p = 2^{32582657} - 1,$$

이다. 이 소수는 프린트하려면, 한 페이지에 한 줄에 80개의 수를 적은 60줄이 프린트될 때, 2000 페이지가 필요하다. The Electronic Frontier Foundation은 10,000,000 자릿수의 소수를 처음으로 찾는 사람에게 $100,000의 상금을 제시하였다. [1]

유클리드의 정리로부터 이 p보다 큰 소수가 무한히 많다는 것을 우리는 안다. 주어진 수가 소수인지 아닌지를 파악하는 것은 이론적으로도 물론이지만, 암호론에도 응용되므로 아주 흥미로운 문제이다. 이 부분은 절 2.4과 단원 3에서 확인할 수 있다.

SAGE 예 1.2.4. Sage에서 p의 10진법 자릿수를 계산할 수 있다. 이를 계산하기 위해서 여러분의 컴퓨터에서는 1분 정도 소요되는 심각한 계산일 수도 있어 조심해야겠지만. 또 너무 긴 시간이 걸릴 수 있기 때문에 아래의 p나 s를 프린트 하지는 말기 바란다.

```
sage: p = 2^32582657 - 1
sage: p.ndigits()
9808358
```

다음 p를 10진법 숫자 열로 변환하고 특정 자릿수를 관찰한다.

```
sage: s = p.str(10) # this takes a long time
sage: len(s)        # s is a very long string  (long time)
9808358
sage: s[:20]        # the first 20 digits of p (long time)
'12457502601536945540'
sage: s[-20:]       # the last 20 digits       (long time)
'11752880154053967871'
```

1.2.4 $ax + b$ 형태의 소수

다음 주제는 $ax + b$ 꼴의 소수이다. 단, a와 b는 $a > 1$인 고정된 정수이고 x는 임의의 자연수 값을 취할 수 있다. $\gcd(a, b) = 1$이라고 가정하자. 왜냐하면, 그렇지 않다면 $ax + b$가 소수가 될 수 있는 희망이 없기 때문이다. 예를 들어 $2x + 2 = 2(x + 1)$는 오직 $x = 0$일 때만 소수이다.

---

[1] 한국어 버전을 수정하고 있는 2019년 7월 현재 가장 큰 소수는 2018년 12월 7일 발견된 자릿수가 24,862,048인 51 번째 메르센 소수로 알려진 $p = 2^{82,589,933} - 1$이다.

기초정리 1.2.5. *$4x-1$꼴의 소수는 무한히 많다.*

왜 이것이 사실일수 있을까? $4x-1$ 꼴의 수를 모두 쓰고 소수인 것에는 밑줄을 그어보자.

$$\underline{3}, \underline{7}, \underline{11}, 15, \underline{19}, \underline{23}, 27, \underline{31}, 35, 39, \underline{43}, \underline{47}, \ldots$$

밑줄 친 수들이 무한히 많이 나올 것 같다고 느껴질 뿐만 아니라, 실제 증명도 쉽다.

증명 $p_1, p_2, \ldots, p_n$은 $4x-1$ 꼴의 서로 다른 소수라고 하자.

$$N = 4p_1p_2\cdots p_n - 1$$

이라고 놓자. 그러면 모든 i에 대하여 $p_i \nmid N$이다. 게다가, $p \mid N$인 모든 소수가 $4x+1$꼴이지는 않다: 만약 그렇다면 N도 $4x+1$꼴이어야만 하는데 N은 $4k-1$꼴이다. 또 N은 홀수이므로, N의 모든 소수 약수도 홀수이다. 따라서 $p \mid N$인 $4x-1$ 꼴의 소수 약수가 존재한다. $p \neq p_i$이므로 $4x-1$ 꼴의 새로운 소수를 찾았다. 이 과정을 무한히 되풀이 할 수 있으므로 $4x-1$ 꼴의 소수는 무한히 많다. □

이 증명은 $4x+1$ 꼴에는 적용할 수 없다. 왜냐하면 $4x-1$ 꼴의 숫자들의 곱은 $4x+1$ 형태가 될 수 있기 때문이다.

예 1.2.6. $p_1 = 3$, $p_2 = 7$이라 하자. 그러면

$$N = 4\cdot 3\cdot 7 - 1 = \underline{83}$$

은 $4x-1$꼴의 소수이다. 다음

$$N = 4\cdot 3\cdot 7\cdot 83 - 1 = \underline{6971},$$

도 $4x-1$꼴의 소수이다. 계속하면

$$N = 4\cdot 3\cdot 7\cdot 83\cdot 6971 - 1 = 48601811 = 61\cdot \underline{796751}$$

이 되는데 이 때 61은 $4x+1 = 4\cdot 15+1$꼴의 소수이지만, 그러나 796751가 $796751 = 4\cdot 199188 - 1$을 만족하는 소수이다. 이와 같이 계속 $4x-1$꼴의 소수를 얻을 수 있다.

$$N = 4\cdot 3\cdot 7\cdot 83\cdot 6971\cdot 796751 - 1 = \underline{5591}\cdot 6926049421.$$

이번에는 작은 소수 5591이 $4x-1$ 꼴의 소수이고 큰 소수가 $4x+1$ 꼴이다.

정리 1.2.7 (Dirichlet)**.** *a와 b가 $\gcd(a,b)=1$를 만족하는 정수이다. 그러면 $ax+b$ 꼴의 소수가 무한히 많이 존재한다.*

이 정리의 증명은 고급 정수론의 전형적인 도구를 사용하므로 이 책의 범위를 넘어선다. (예를 들어 [18, §VIII.4]를 참조.)

1.2.5 소수의 밀도

절 1.2.1에서 소수가 무한히 많이 있음을 알았다. 이 절에서는 많은 소수들이 어떻게 존재하는지를 알기 위하여 몇 가지 준비운동격인 질문들을 생각해본다. 그리고 몇 가지 수치적 증거들을 소개하고, 우리의 질문에 대한 점근적인 답을 주는 소수정리(prime number theorem)를 기술한다. 그리고 이 정리와 그 유명한 리만 가설(Riemann Hypothesis)과의 연관성을 알아본다. 이 절에서 소수를 세는 우리의 토론 방식은 아주 피상적이다: 자세한 내용은 Crandall과 Pomerance의 멋진 책 [12, §1.1.5]을 읽기 바란다.

소수의 수 (혹은 퍼센트)를 측정하는 정확한 방법을 설명하기 위하여 다음 질문들을 생각해보자. 자연수의 몇 %가 짝수인가? 그 답은 50%이다. 자연수 중의 몇 %가 $4x-1$의 꼴인가? 답: 자연수의 1/4, 혹은 25%. 자연수 중의 몇 %가 완전제곱수인가? 답: 0 퍼센트! 전체 자연수에서 제곱수가 차지하는 비율이 결국은 0에 수렴한다는 의미에서다. 좀 더 정확하게 설명하면,

$$\lim_{x\to\infty} \frac{\#\{n \in \mathbf{N} : n \le x \text{ and } n \text{ is a perfect square}\}}{x} = 0$$

이다. 왜냐하면 위의 식에서 분자는 대략 $\sqrt{x}$이고 $\lim_{x\to\infty} \frac{\sqrt{x}}{x} = 0$이기 때문이다. 비슷하게 정리 1.2.10으로부터 모든 자연수에서 소수는 0 퍼센트를 차지함을 알 수 있다. (see Exercise 1.4).

따라서 우리는 다음과 같은 다른 질문을 하게 한다: x보다 작거나 같은 수 중에서 제곱수는 모두 몇 개인가? 답: 약 $\sqrt{x}$. 다음은 소수에 대한 유사한 질문이다.

질문 1.2.8. x보다 작거나 같은 수 중에서 소수는 몇 개인가?

$$\pi(x) = \#\{p \in \mathbf{N} : p \le x,\ p \text{ 는 소수}\}$$

라고 하자. 그러면

$$\pi(6) = \#\{2, 3, 5\} = 3.$$

이다. 표 1.1에 $\pi(x)$의 값들이 주어져있다. 그림 1.1과 1.2는 $\pi(x)$의 그래프들이다. 이 그래프들은 직선처럼 보이지만 약간은 아래로 휘어져 있다.

SAGE 예 1.2.9. Sage에서 `prime_pi(x)` 명령어를 사용하면 $\pi(x)$를 구할 수 있다.

```
sage: prime_pi(6)
3
sage: prime_pi(100)
25
sage: prime_pi(3000000)
216816
```

`plot` 명령어를 써서 $\pi(x)$의 대략적인 모습을 그릴 수 있다.

```
sage: plot(prime_pi, 1,1000, rgbcolor=(0,0,1))
```

TABLE 1.1. Values of $\pi(x)$

x	100	200	300	400	500	600	700	800	900	1000
$\pi(x)$	25	46	62	78	95	109	125	139	154	168

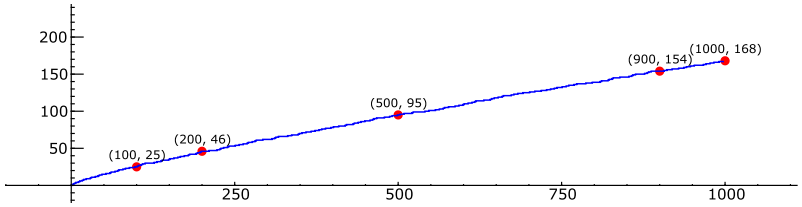


FIGURE 1.1. Graph of $\pi(x)$ for $x < 1000$

Gauss는 거의 컴퓨터와 같았는데, 1849년 편지에 $3,000,000$ 보다 작은 수 중에는 $216,745$ 개의 소수가 존재한다고 편지에 쓸 정도였다. (Gauss의 이 계산 값은 비슷하기는 했지만 정확하지는 않았다. 실제 값은 $216,816$이다.)

Gauss는 다음에 소개할 $\pi(x)$에 대한 근사 공식을 추론하였는데, 1896년에 Hadamard와 Vallée Poussin이 독립적으로 증명하였다. (물론 이 책에서는 증명하지 않는다.)

정리 1.2.10 (소수정리)**.** *$\pi(x)$와 $x/\log(x)$는*

$$\lim_{x\to\infty} \frac{\pi(x)}{x/\log(x)} = 1$$

을 만족한다. 이를 우리는 $\pi(x)$는 $x/\log(x)$에 점근(asymptotic)한다고 한다. 즉 $\pi(x)$의 점근곡선이 $x/\log(x)$이다.

이 정리에 대하여 관찰을 좀 더 해보자. 이 정리는

$$\lim_{x\to\infty} \frac{\pi(x)}{x} = \lim_{x\to\infty} \frac{1}{\log(x)} = 0$$

을 유도하므로, 임의의 a에 대하여

$$\lim_{x\to\infty} \frac{\pi(x)}{x/(\log(x)-a)} = \lim_{x\to\infty} \frac{\pi(x)}{x/\log(x)} - \frac{a\pi(x)}{x} = 1.$$

따라서 $x/(\log(x)-a)$ 또한 $\pi(x)$의 점근식이다. 왜 $a=1$이 최선의 선택인지는 [12, §1.1.5]를 참고하기 바란다. 표 1.2 는 $x < 10000$ 인 여러 값에서 $\pi(x)$와 $x/(\log(x)-1)$를 비교한다.

다음은 이 책을 서술하는 당시의 $\pi(x)$의 최고 기록이다.

$$\pi(10^{23}) = 1925320391606803968923.$$

사실 이런 계산들은 정확한 값을 얻는 것이 아주 어렵기 때문에 위의 값이 약간은 틀릴 수 있음을 지적하고 싶다.

TABLE 1.2. Comparison of $\pi(x)$ and $x/(\log(x)-1)$

x	$\pi(x)$	$x/(\log(x)-1)$ (approx)
1000	168	169.269029060440816518625 6278
2000	303	302.988873454546387802980 0994
3000	430	428.181931797523704374738 5740
4000	550	548.392209727825326413340 0985
5000	669	665.141878448650217236945 5815
6000	783	779.269888585477862686367 7374
7000	900	891.303565722333997435256 7759
8000	1007	1001.602962794770080754784281
9000	1117	1110.428422963188172310675011
10000	1229	1217.976301461550279200775705

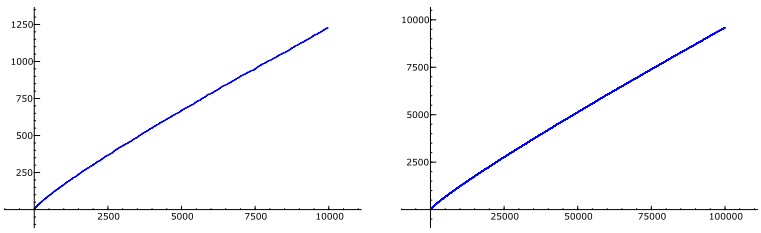


FIGURE 1.2. Graphs of $\pi(x)$ for $x < 10000$ and $x < 100000$

복소해석학에 익숙한 독자들을 위하여 $\pi(x)$와 리만가설과의 관계를 언급한다. 리만제타함수 $\zeta(s)$는 오른쪽 반평면 위에서 $\sum_{n=1}^{\infty} n^{-s}$로 정의된 함수를 $\mathbf{C}\setminus\{1\}$로 확장한 복소해석함수이다. 리만 가설은 실수 부분이 양수인 복소수 중에서 $\zeta(s)$가 영이 되는 복소수는 모두 $\mathrm{Re}(s) = 1/2$인 직선위에 놓여 있다는 가설이다. 이 가설은 Clay 수학연구소의 백만불짜리 새 천년 문제 중의 하나이다 [8].

[12, §1.4.1]에 따르면, 리만가설은

$$\mathrm{Li}(x) = \int_2^x \frac{1}{\log(t)} dt$$

가 $\pi(x)$의 '좋은' 근삿값이라는 가설과 동치라고 알려져 있다. 정확하게 설명하면 다음과 같다.

가설 1.2.11 (리만가설과 동치인)**.**
$x \geq 2.01$인 모든 x에 대하여,

$$|\pi(x) - \mathrm{Li}(x)| \leq \sqrt{x}\log(x).$$

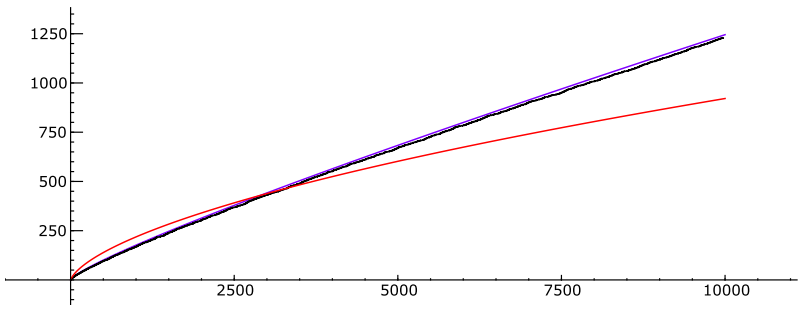
만약 $x = 2$이면 $\pi(2) = 1$, $\mathrm{Li}(2) = 0$이다. 그러나 $\sqrt{2}\log(2) = 0.9802\ldots$ 이므로 위의 부등식은 $x \geq 2$에 대해서는 성립하지 않지만 2.01이면 충분하다. 이 가설은 더 설명은 하지 않고 수치적 예를 제시하고 마무리한다.

예 1.2.12. $x = 4 \cdot 10^{22}$리고 하자. 그러면

$$\begin{aligned}
\pi(x) &= 783964159847056303858,\\
\mathrm{Li}(x) &= 783964159852157952242.7155276025801473\ldots,\\
|\pi(x) - \mathrm{Li}(x)| &= 5101648384.71552760258014\ldots,\\
\sqrt{x}\log(x) &= 10408633281397.77913344605\ldots,\\
x/(\log(x) - 1) &= 783650443647303761503.5237113087392967\ldots.
\end{aligned}$$

SAGE 예 1.2.13. Sage을 시용하여 $\pi(x)$, $\mathrm{Li}(x)$, 그리고 $\sqrt{x}\log(x)$를 그려보자.

```
sage: P = plot(Li, 2,10000, rgbcolor='purple')
sage: Q = plot(prime_pi, 2,10000, rgbcolor='black')
sage: R = plot(sqrt(x)*log(x),2,10000,rgbcolor='red')
sage: show(P+Q+R,xmin=0, figsize=[8,3])
```



가장 위의 선은 $\mathrm{Li}(x)$, 다음은 $\pi(x)$, 가장 아래 선은 $\sqrt{x}\log(x)$이다.

소수정리(prime number theorem)와 리만 가설(Riemann hypothesis)은 [55]와 [38]를 참조한다.

1.3 Exercises

1.1 Compute the greatest common divisor $\gcd(455, 1235)$ by hand.

1.2 Use the prime enumeration sieve to make a list of all primes up to 100.

1.3 Prove that there are infinitely many primes of the form $6x - 1$.

1.4 Use Theorem 1.2.10 to deduce that $\lim_{x\to\infty} \dfrac{\pi(x)}{x} = 0$.

1.5 Let $\psi(x)$ be the number of primes of the form $4k-1$ that are $\leq x$. Use a computer to make a conjectural guess about $\lim_{x\to\infty} \psi(x)/\pi(x)$.

1.6 So far 44 Mersenne primes $2^p - 1$ have been discovered. Give a guess, backed up by an argument, about when the next Mersenne prime might be discovered (you will have to do some online research).

1.7 (a) Let $y = 10000$. Compute $\pi(y) = \#\{\text{primes } p \leq y\}$.

(b) The prime number theorem implies $\pi(x)$ is asymptotic to $\frac{x}{\log(x)}$. How close is $\pi(y)$ to $y/\log(y)$, where y is as in (a)?

1.8 Let a, b, c, n be integers. Prove that

(a) if $a \mid n$ and $b \mid n$ with $\gcd(a, b) = 1$, then $ab \mid n$.

(b) if $a \mid bc$ and $\gcd(a, b) = 1$, then $a \mid c$.

1.9 Let a, b, c, d, and m be integers. Prove that

(a) if $a \mid b$ and $b \mid c$ then $a \mid c$.

(b) if $a \mid b$ and $c \mid d$ then $ac \mid bd$.

(c) if $m \neq 0$, then $a \mid b$ if and only if $ma \mid mb$.

(d) if $d \mid a$ and $a \neq 0$, then $|d| \leq |a|$.

1.10 In each of the following, apply the division algorithm to find q and r such that $a = bq + r$ and $0 \leq r < |b|$:

$$a = 300, b = 17,\ \ a = 729, b = 31,\ \ a = 300, b = -17,\ \ a = 389, b = 4.$$

1.11 (a) (Do this part by hand.) Compute the greatest common divisor of 323 and 437 using the algorithm described in class that involves quotients and remainders (i.e., do not just factor a and b).

(b) Compute by any means the greatest common divisor of

$$314159265358979323846264338$$

and

$$271828182845904523536028747.$$

1.12 (a) Suppose a, b and n are positive integers. Prove that if $a^n \mid b^n$, then $a \mid b$.

(b) Suppose p is a prime and a and k are positive integers. Prove that if $p \mid a^k$, then $p^k \mid a^k$.

1.13 (a) Prove that if a positive integer n is a perfect square, then n cannot be written in the form $4k + 3$ for k an integer. (Hint: Compute the remainder upon division by 4 of each of $(4m)^2$, $(4m + 1)^2$, $(4m + 2)^2$, and $(4m + 3)^2$.)

(b) Prove that no integer in the sequence

$$11, 111, 1111, 11111, 111111, \ldots$$

is a perfect square. (Hint: $111 \cdots 111 = 111 \cdots 108 + 3 = 4k + 3$.)

1.14 Prove that a positive integer n is prime if and only if n is not divisible by any prime p with $1 < p \leq \sqrt{n}$.

2

법 n 정수들의 집합 (The ring of Integers Modulo n)

정말 놀라운 사실은 1000 자릿수의 *인수를 실제 찾아내지도 못하면서* 그 수가 소수인지 아닌지를 판정하는데는 일초도 채 걸리지 않는다는 것이다. 이와 같이 소수인지 아닌지를 판정하는 알고리듬은 n이 소수냐 아니냐에 따라 달라지는 n을 기준으로 한 어떤 연산의 성질에 의존한다. 실제 우리는 정수들의 집합 $\{0, 1, \ldots, n-1\}$에 속하는 두 원소를 더하거나 곱해도 다시 이 속에 들어가는 혁신적인 덧셈과 곱셈을 정의하고 여러 재미있는 연산 규칙들을 찾아낸다.

또 다른 놀라운 사실은 $n = 1234^{1234567890}$와 같이 어마어마하게 큰 수를 다 계산할 필요없이 마지막 1000자릿수를 즉석에서 계산할 수 있다. 물론 이것도 집합 $\{0, 1, \ldots, n-1\}$의 연산에 의존한다.

이 단원은 곧 정의할 법 n 연산으로 환이라는 대수적 구조를 갖는 $\mathbf{Z}/n\mathbf{Z}$에 관한 단원으로, 이 단원에서는 이 환의 멋진 구조와 그 구조를 위에서 소개한 문제들과 또 다른 문제들에 어떻게 응용하는지를 소개한다. 이 단원은 이 책의 나머지에서 기초가 된다. 2.1절에서는 법 n에서의 일차방정식이 해를 갖는지를 소개한 후 Euler의 φ 함수를 소개하고 Euler의 정리와 Wilson의 정리를 증명한다. 2.2절에서는 중국인의 나머지 정리를 증명한다. 중국인의 나머지 정리란 각 식들의 법이 서로 소일 때 연립일차합동식의 해에 관한 정리이다. 이런 이론들을 기초로 하여 2.3절에서는 법 n에서의 거듭제곱과 일차합동식의 해를 구하는 알고리듬을 포함하여 강력한 계산 알고리듬을 소개한다. 마지막으로 2.4절에서 법 n에서의 산술을 이용하여 소수를 판정하는 방법을 논한다.

2.1 법 n 합동

정의 2.1.1 (군)**. 군(group)**은 다음 성질을 만족하는 항등원 1이 있는 이항연산[1] $G \times G \to G$을 갖는 집합이다.

1. 모든 $a, b, c \in G$에 대하여 $(ab)c = a(bc)$가 성립;
2. G의 각의 원소 a에 대하여 $1a = a1 = a$가 성립하고, $ab = 1$인 $b \in G$가 존재.

정의 2.1.2 (아벨군)**.** 아벨군(abelian group) G는 모든 $a, b \in G$가 $ab = ba$를 만족하는 군이다.

정의 2.1.3 (환)**. 환(ring)** R은 두 개의 이항연산 $+$와 $\times$를 갖는 집합으로 덧셈 $+$에 관해서는 항등원 0을 갖는 아벨군이고 곱셈에 관해서는 모든 $a, b, c \in R$에 대하여

- $1a = a1 = a$인 $1 \in R$이 존재
- $(ab)c = a(bc)$
- $a(b + c) = ab + ac, \ \ (a + b)c = ac + bc$

을 만족하는 집합이다. 만약 모든 $a, b \in R$에 대하여 $ab = ba$를 추가로 만족하면 R은 **가환환(commutative ring)**이다.

이 절에서는 $\mathbf{Z}/n\mathbf{Z}$으로 표현할 법 n 정수들의 환을 정의하고, $\mathbf{Z}/n\mathbf{Z}$의 원소의 곱셈위수와 관련이 되는 Euler φ-함수를 소개한다.

$n \in \mathbf{N}$이고 $a, b \in \mathbf{Z}$일 때, $n \mid a - b$가 성립하면 a**와** b**는 법** n **합동**이라고 하고 $a \equiv b \pmod{n}$로 쓴다. $a \equiv b \pmod{n}$는 정수들의 집합 $\mathbf{Z}$에서 동치관계가 된다. n의 배수들의 집합을 $n\mathbf{Z} = (n)$로 쓰자. (집합 $n\mathbf{Z} = (n)$는 n으로 생성된 $\mathbf{Z}$의 아이디얼(ideal)이라 불리는 집합이다.)

정의 2.1.4 (법 n 정수)**. 법** n **정수(integers modulo** n**)**들의 환 $\mathbf{Z}/n\mathbf{Z}$은 법 n 합동(congruent modulo n)에서 동치류들의 집합이다. 이 집합에 자연스럽게 다음과 같이 덧셈과 곱셈을 정의하면 환이 된다.

$$(a + n\mathbf{Z}) + (b + n\mathbf{Z}) = (a + b) + n\mathbf{Z}$$

$$(a + n\mathbf{Z}) \cdot (b + n\mathbf{Z}) = (a \cdot b) + n\mathbf{Z}.$$

예 2.1.5. 예를 들어

$$\mathbf{Z}/3\mathbf{Z} = \{\{\ldots, -3, 0, 3, \ldots\}, \{\ldots, -2, 1, 4, \ldots\}, \{\ldots, -1, 2, 5, \ldots\}\}.$$

SAGE 예 2.1.6. Sage에서는 다음과 같이 $\mathbf{Z}/n\mathbf{Z}$의 원소를 다 나열할 수 있다.

---

[1] 집합 X의 이항연산이란 X의 두 원소를 연산하여 다시 X에 속하는 연산이다. 정수집합에서 덧셈과 뺄셈은 이항연산이다.

```
sage: R = Integers(3)
sage: list(R)
[0, 1, 2]
```

법 n에 관한 잉여류들의 집합을 $\mathbf{Z}/n\mathbf{Z}$으로 표현하는 이유는 $\mathbf{Z}/n\mathbf{Z}$는 정수들의 환 $\mathbf{Z}$를 n의 배수들의 집합인 $n\mathbf{Z}$로 나누어서 얻은 몫과 같은 대상이기 때문이다. 이 때문에 $\mathbf{Z}$의 덧셈과 곱셈이 $\mathbf{Z}/n\mathbf{Z}$의 덧셈과 곱셈을 자연스럽게 유도한다. 법 n에서 a의 동치류 $a+n\mathbf{Z}$를 간단히 a(Sage 예 2.1.6의 출력 참조), 혹은 $a \pmod n$으로 표기한다.

정의 2.1.7 (체)**. 체(field)** K는 K의 0이 아닌 모든 원소 a가 $ab = 1$을 만족하는 b를 K안에 가지고 있는 가환환이다.

예를 들어 p가 소수이면, $\mathbf{Z}/p\mathbf{Z}$는 체이다. (Exercise 2.12 참조).

정의 2.1.8 (축약 함수와 올림)**.** 정수 a를 $a + n\mathbf{Z}$로 보내는 자연스러운 함수 $\gamma : \mathbf{Z} \to \mathbf{Z}/n\mathbf{Z}$를 **법** n **축약(reduction modulo** n**)**이라 부른다. $\gamma(b) = a + n\mathbf{Z}$인 b를 $a + n\mathbf{Z}$의 법 n **올림(lift)**이라 한다. 따라서, $7 + 3\mathbf{Z} = 1 + 3\mathbf{Z}$ 이므로, 7은 $1 \pmod 3$의 올림이다.

어떤 수가 n으로 나누어지는지를 확인하기 위하여도 $\mathbf{Z}/n\mathbf{Z}$에서의 연산이 잘 정의되었다는 사실을 이용할 수 있다. (Exercise 2.8 참조)

기초정리 2.1.9. *정수 $n \in \mathbf{Z}$이 3으로 나누어떨어지기 위한 필요충분조건은 n의 자릿수의 합이 3으로 나누어떨어지는 것이다.*

증명 정수 n의 자릿수가 a, b, c, 등으로 계속된다면

$$n = a + 10b + 100c + \cdots$$

으로 쓸 수 있다. $10 \equiv 1 \pmod 3$, $100 \equiv 1 \pmod 3$,

$$n = a + 10b + 100c + \cdots \equiv a + b + c + \cdots \pmod 3,$$

이 성립하고 이 사실로부터 우리의 주장은 증명된다. □

2.1.1 일차합동식

이 절에서는 일차합동식 $ax \equiv b \pmod n$가 법 n에서 해을 가지는지 아닌지를 결정할 수 있는 방법을 찾으려고 한다. $ax \equiv b \pmod n$의 해를 *계산하는* 알고리듬은 2.3절의 주제이다.

먼저 양변에 같은 수가 곱해져 있는 합동식에서 언제 소거할 수 있는지를 알아본다.

기초정리 2.1.10 (소거)**.** *만약* $\gcd(c, n) = 1$*이고*

$$ac \equiv bc \pmod n$$

이면 $a \equiv b \pmod n$*이 성립한다.*

증명 정의에 의하여

$$n \mid ac - bc = (a-b)c.$$

$\gcd(n,c) = 1$이므로 정리 1.1.6 (혹은 연습문제 1.8(b))으로부터 $n \mid a-b$이다. 따라서

$$a \equiv b \pmod{n}$$

를 얻는다. □

$\mathbf{Z}/n\mathbf{Z}$에서 a'이 a의 곱셈역원이면 (즉, $aa' \equiv 1 \pmod{n}$이면) 합동식 $ax \equiv b \pmod{n}$는 유일한 해 $x \equiv a'b \pmod{n}$를 갖는다. 따라서 $\mathbf{Z}/n\mathbf{Z}$에서 곱셈역원을 갖는 모든 원소들을 다 결정할 수 있다면 아주 흥미로울 것이다. 곱셈역원을 갖는 원소를 **가역원(invertible element, unit)**이라 한다.

이제 법 n 완전잉여집합을 이용하여 $a \in \mathbf{Z}/n\mathbf{Z}$가 $\mathbf{Z}/n\mathbf{Z}$의 가역원일 조건은 a의 임의로 잡은 올림 $\tilde{a}$이 $\gcd(\tilde{a}, n) = 1$임을 보일려고 한다.

정의 2.1.11 (완전잉여집합)**.** 원소가 n개인 $\mathbf{Z}$의 부분집합 R의 모든 원소가 법 n에서 서로 합동이 아니면 집합 R은 법 n의 **완전잉여집합(complete set of residues)**이다. 즉, $\mathbf{Z}/n\mathbf{Z}$의 각 원소에서 정확히 한 개의 대표들만을 뽑아서 구성한 집합이 완전잉여집합이다.

예를 들어

$$R = \{0, 1, 2, \ldots, n-1\}$$

은 법 n의 완전잉여집합이다. $n = 5$이면, $R = \{0, 1, -1, 2, -2\}$, 혹은 $\{5, 6, 7, 8, 9\}$ 등이 완전잉여집합이다.

보조정리 2.1.12. *집합 R이 법 n의 완전잉여집합이다. $a \in \mathbf{Z}$가 $\gcd(a,n) = 1$를 만족하면 $aR = \{ax : x \in R\}$도 법 n의 완전잉여집합이다.*

증명 $x, x' \in R$이고 $ax \equiv ax' \pmod{n}$이면 기초정리 2.1.10 에 의하여 $x \equiv x' \pmod{n}$가 성립한다. R이 완전잉여집합이므로 $x = x'$이다. 따라서 aR의 모든 원소는 법 n의 다른 잉여류들의 대표이다. 그런데 $\#aR = n$이므로 aR은 법 n의 완전잉여집합이다. □

기초정리 2.1.13 (가역원)**.** $\gcd(a,n) = 1$*이면* $ax \equiv b \pmod{n}$ *는 해를 갖고 또 그 해는 법 n에 관하여 유일하다.*

증명 법 n의 완전잉여집합을 하나 잡고 그것을 R이라고 하자. 보조정리 2.1.12 에 의하여 aR도 법 n의 완정잉여집합이다. 따라서 aR에는 b와 합동인 원소가 딱 하나 존재한다. 이 원소를 $ax \in aR$라고 쓰면 x는 우리가 찾는 $ax \equiv b \pmod{n}$의 해이며 유일하다. □

대수적으로 이 기초정리는 $\gcd(a,n) = 1$이면, a를 왼쪽에 곱해주는 함수 $\mathbf{Z}/n\mathbf{Z} \to \mathbf{Z}/n\mathbf{Z}$는 일대일 대응이라는 주장과 동일하다.

예 2.1.14. 일차합동식 $2x \equiv 3 \pmod 7$를 생각하고 위의 기초정리의 증명과정을 따라가보자. $R = \{0,1,2,3,4,5,6\}$은 법 7의 완전잉여집합이고, $\gcd(2,7) = 1$이다. 그러면

$$2R = \{0, 2, 4, 6, 8 \equiv 1, 10 \equiv 3, 12 \equiv 5\}$$

은 다시 완전잉여집합가 된다. 또 $3 \in 2R$ 이므로 $2 \cdot 5 \equiv 3 \pmod 7$를 얻는다. 따라서 $2x \equiv 3 \pmod 7$의 해는 $x = 5$이다.

$\gcd(a,n) \neq 1$이면 방정식 $ax \equiv b \pmod n$은 해를 가질 수도 갖지 않을 수도 있다. 예를 들어 $2x \equiv 1 \pmod 4$는 해가 없지만, $2x \equiv 2 \pmod 4$는 해를 가지는데 법 4에 관하여 한 개 이상의 해를 가진다. 실제 $x = 1$과 $x = 3$이 모두 해이다. 기초정리 2.1.13을 일반화하여 일차 합동식의 해의 존재에 대한 다음 기준을 얻는다.

기초정리 2.1.15 (일차합동식의 해의 존재성)**.** *일차 합동식 $ax \equiv b \pmod n$이 해를 가질 필요충분조건은 $\gcd(a,n)$이 b를 나누는 것이다.*

증명 $g = \gcd(a,n)$이라 하자. $ax \equiv b \pmod n$의 해 x가 존재한다고 하자. 그러면 당연히 $n \mid (ax - b)$이다. 그런데 $g \mid n$, $g \mid a$이므로 $g \mid b$가 성립한다.

역으로 $g \mid b$라고 하자. 그러면 $n \mid (ax - b)$는

$$\frac{n}{g} \mid \left(\frac{a}{g}x - \frac{b}{g}\right)$$

와 동치이다. 따라서 $ax \equiv b \pmod n$가 해를 갖는다는 것과 $\frac{a}{g}x \equiv \frac{b}{g} \pmod{\frac{n}{g}}$가 해를 갖는다는 것이 동치이다. $\gcd(a/g, n/g) = 1$이므로, 기초정리 2.1.13에 의하여 마지막 식이 해을 가지므로 증명이 완성된다. □

이제 일차합동식이 해를 갖는 판정법을 알았다. 단원 4에서는 법 n의 이차합동식이 해를 갖는지를 판정할 수 있는 이차상호법칙을 공부할 것이다.

2.1.2 Euler 정리

$(\mathbf{Z}/n\mathbf{Z})^*$는 $\gcd(x,n) = 1$인 $[x] \in \mathbf{Z}/n\mathbf{Z}$들의 집합을 의미한다. ($[x]$는 $x + n\mathbf{Z}$이다.)

우리에게 아주 흥미로운 이 집합은 군이 되는데 이 군은 환 $\mathbf{Z}/n\mathbf{Z}$의 가역원들의 군이라고도 불린다. 이 군의 모든 원소는 $(\mathbf{Z}/n\mathbf{Z})^*$의 원소의 개수만큼 거듭제곱하면 항상 1이 된다. 군론에서 나오는 Lagrange정리와 연관된 성질인데, 우리는 n이 소수인 경우는 "Fermat's Little Theorem"로, 일반적인 경우는 "Euler's Theorem"라는 이름으로 알려져 있는 이 정리를 군론의 이론을 쓰지 않고 이 절에 소개하는 기본 원칙으로 증명할 것이다.

정의 2.1.16 (위수)**.** $n \in \mathbf{N}$, $x \in \mathbf{Z}$이고 $\gcd(x,n) = 1$이라 하자. x의 **법** n **위수(order)**는

$$x^m \equiv 1 \pmod n$$

이 되는 작은 자연수 $m \in \mathbf{N}$이다.

이 정의가 의미가 있기 위해서는 그런 m이 존재함을 보여야 한다. 법 n에서 $x, x^2, x^3, \ldots$을 계산하자. 법 n 잉여류들은 오직 유한개 뿐이므로

$$x^j \equiv x^i \pmod{n}.$$

인 i와 j가 존재할수 밖에 없다. $\gcd(x, n) = 1$이므로, 기초정리 2.1.10으로부터 x^i를 소거할 수 있으므로 ($i < j$를 가정하면)

$$x^{j-i} \equiv 1 \pmod{n}.$$

를 얻는다.

SAGE 예 2.1.17. Sage에서 `x.multiplicative_order()`를 이용하여 $\mathbf{Z}/n\mathbf{Z}$의 원소의 위수를 구하자.

```
sage: R = Integers(10)
sage: a = R(3)                  # create an element of Z/10Z
sage: a.multiplicative_order()
4
```

원소 a의 위수가 4라는 것은 a의 거듭제곱들이 주기가 4인 순환수열을 이룬다는 것이다. 다음은 거듭제곱들을 계산하라는 코드이다.

```
sage: [a^i for i in range(15)]
[1, 3, 9, 7, 1, 3, 9, 7, 1, 3, 9, 7, 1, 3, 9]
```

명령어 `range(n)`는 0 and $n-1$까지의 정수 범위에서 주어진 값을 구하라는 명령어이다.

정의 2.1.18 (Euler의 φ-함수)**.** 자연수 n에 대하여 n과 서로 소인 n이하의 자연수들의 개수를 $\varphi(n)$이라 하고 Euler φ-함수라고 한다. 즉,

$$\varphi(n) = \#\{a \in \mathbf{N} : a \leq n \text{ and } \gcd(a, n) = 1\}.$$

예를 들어,

$$\begin{aligned}\varphi(1) &= \#\{1\} = 1,\\ \varphi(2) &= \#\{1\} = 1,\\ \varphi(5) &= \#\{1, 2, 3, 4\} = 4,\\ \varphi(12) &= \#\{1, 5, 7, 11\} = 4.\end{aligned}$$

p가 소수이면

$$\varphi(p) = \#\{1, 2, \ldots, p-1\} = p - 1.$$

절2.2.1에서 $\gcd(m, r) = 1$이면 $\varphi(mr) = \varphi(m)\varphi(r)$임을 증명한다. 이를 이용하면 n의 소인수분해로부터 $\varphi(n)$을 쉽게 계산할 수 있다.

SAGE 예 2.1.19. `euler_phi(n)`은 $\varphi(n)$을 계산하는 Sage코드이다.

```
sage: euler_phi(2007)
1332
```

정리 2.1.20 (Euler 정리)**.** $\gcd(x,n)=1$*이면*

$$x^{\varphi(n)} \equiv 1 \pmod{n}$$

이 성립한다.

증명 위에서 지적하였듯이 Euler의 정리는 군의 이론을 이용하여 설명할 수도 있다. 먼저 $\mathbf{Z}/n\mathbf{Z}$의 가역원들의 집합

$$(\mathbf{Z}/n\mathbf{Z})^* = \{a \in \mathbf{Z}/n\mathbf{Z} : \gcd(a,n)=1\}$$

은 위수가 $\varphi(n)$인 군이다. 그러면 위의 정리는 $(\mathbf{Z}/n\mathbf{Z})^*$의 원소들의 위수는 $(\mathbf{Z}/n\mathbf{Z})^*$의 위수인 $\varphi(n)$을 나눈다는 것이다. 이는 유한군에 관한 Lagrange의 정리의 특별한 경우이다. Lagrange의 정리: G가 유한군이고 $g \in G$ 이면 g의 위수는 군 G의 위수를 나눈다. 여기에서는 군론의 성질을 이용하지 않고 증명한다.

$$P = \{a : 1 \le a \le n \text{ and } \gcd(a,n)=1\}$$

라 하면 보조정리 2.1.12를 증명했던 똑 같은 방법으로 법 n에 관해서 비교하면 P와 xP는 같은 집합이 된다. 따라서 P의 모든 원소들의 곱과 xP의 모든 원소들의 곱이 법 n에서는 같다.

$$\prod_{a\in P}(xa) \equiv \prod_{a\in P} a \pmod{n}.$$

양 변의 a들을 모두 소거하면 주장한대로

$$x^{\#P} \equiv 1 \pmod{n}$$

을 얻는다. □

SAGE 예 2.1.21. Sage를 사용하여 오일러의 정리를 예로 확인한다. 임의의 정수 x에 대하여 Sage코드 `Mod(x,n)`는 $\mathbf{Z}/n\mathbf{Z}$에 속하는 x의 동치류를 계산한다.

```
sage: n = 20
sage: k = euler_phi(n); k
8
sage: [Mod(x,n)^k for x in range(n) if gcd(x,n) == 1]
[1, 1, 1, 1, 1, 1, 1, 1]
```

2.1.3 Wilson의 정리

다음은 1770년대부터 윌슨의 정리로 알려진 소수의 특성이다. 그런데 이 정리를 최초로 증명한 사람은 Lagrange이다.

기초정리 2.1.22 (윌슨의 정리)**.** *$p > 1$가 소수일 필요충분조건은 $(p-1)! \equiv -1 \pmod{p}$이다.*

예를 들어, $p = 3$이면 $(p-1)! = 2 \equiv -1 \pmod{3}$. $p = 17$이면

$$(p-1)! = 20922789888000 \equiv -1 \pmod{17}.$$

그러나 $p = 15$이면

$$(p-1)! = 87178291200 \equiv 0 \pmod{15}$$

이므로 15는 합성수이다. 따라서 윌슨의 정리는 소수판정 정리로 볼 수 있으나 계산의 관점에서 볼 때 세상에서 가장 비효율적인 소수판정법이다. 왜냐하면 $(n-1)!$를 계산하는 것은 아주 복잡하기 때문이다.

증명 $p = 2$인 경우는 자명하므로, 지금부터는 $p > 2$라고 가정하자. 먼저 p는 소수라고 가정하고 $(p-1)! \equiv -1 \pmod{p}$임을 증명하자. $a \in \{1, 2, \ldots, p-1\}$일 때, $\gcd(a, p) = 1$이므로

$$ax \equiv 1 \pmod{p}$$

은 유일한 해 $a' \in \{1, 2, \ldots, p-1\}$를 가짐을 기억하자. 만약 $a = a'$이면 $a^2 \equiv 1 \pmod{p}$을 만족하므로 $p \mid a^2 - 1 = (a-1)(a+1)$이다. 따라서 $p \mid (a-1)$이거나 $p \mid (a+1)$이므로 $a \in \{1, p-1\}$. 즉, $\{1, 2, \ldots, p-1\}$에서 1고 $p-1$을 제외한 $\{2, 3, \ldots, p-2\}$의 모든 원소들을 각각 자신과 다른 자신의 역원들과 짝을 이루므로, 모두 곱하면

$$2 \cdot 3 \cdot \cdots \cdot (p-2) \equiv 1 \pmod{p}.$$

을 얻는다. 여기에 양변에 $p-1$를 곱하면 $(p-1)! \equiv -1 \pmod{p}$이 증명된다.

다음, $(p-1)! \equiv -1 \pmod{p}$를 만족하면 p 는 소수여야만 함을 증명하자. 이를 위해 소수가 아니라고 가정하자. 그러면 p는 4이상의 합성수이다. ℓ을 p의 소수 약수라고 하자. 그러면 $\ell < p$이므로 당연히 $\ell \mid (p-1)!$. 또 가정에 의하여

$$\ell \mid p \mid ((p-1)! + 1)$$

이므로 $\ell \mid 1$이 성립해야 하는데 이는 모순이다. □

예 2.1.23. 윌슨의 정리의 증명의 핵심 단계인 $\{2, 3, \ldots, p-2\}$의 모든 원소들이 각각 자신의 역원들과 짝을 완전히 이루어 나갈 수 있음을 $p = 17$일 때 확인하자.

$$2 \cdot 3 \cdots 15 = (2 \cdot 9) \cdot (3 \cdot 6) \cdot (4 \cdot 13) \cdot (5 \cdot 7) \cdot (8 \cdot 15) \cdot (10 \cdot 12) \cdot (14 \cdot 11) \equiv 1 \pmod{17}.$$

SAGE 예 2.1.24. Sage를 이용하여 n, $(n-1)!\pmod n$, $-1\pmod n$으로 구성된 세 쌍의 표를 만들어 보자. 아래 표에서 첫 번째 열은 n, 두 번째 열은 $(n-1)!\pmod n$, 세 번째 열은 $-1\pmod n$이다. 첫 번째 열이 소수인 경우에만 두번째와 세 번째 열이 같음을 확인할 수 있다. (Sage에서 ...은 여러 줄로 나타나는 것을 표시하는 것으로 점들을 꼭 타이핑할 필요는 없다.)

```
sage: for n in range(1,10):
...     print n, Mod(factorial(n-1), n), Mod(-1, n)
1 0 0
2 1 1
3 2 2
4 2 3
5 4 4
6 0 5
7 6 6
8 0 7
9 0 8
```

2.2 중국인의 나머지 정리

이 절에서는 중국인의 나머지 정리로 알려진 연립 일차합동식이 해를 가질 조건을 증명한다. 4세기에 살았던 중국의 한 수학자가 다음과 같은 질문을 하였다.

질문 2.2.1. 3개씩으로 나누면 2개가 남고, 5개씩으로 나누면 세 개가 남고, 7개씩으로 나누면 두 개가 남는다면 이 양은 얼마나 되는 것인가?

중국인의 이 질문을 현대 수학으로 표현하면 다음 세 개의 일차 합동식

$$\begin{aligned} x &\equiv 2 \pmod 3 \\ x &\equiv 3 \pmod 5 \\ x &\equiv 2 \pmod 7 \end{aligned}$$

을 동시에 만족하는 양의 정수를 찾는 것이다. 중국인의 나머지 정리(Chinese Remainder Theorem)는 해가 존재한다는 것이고 이 해를 찾는 한 가지 방법을 증명에서 볼 수 있다. (필요한 알고리듬은 절 2.3을 참조)

정리 2.2.2 (중국인의 나머지 정리)**.** *$a, b \in \mathbf{Z}$이고 $n, m \in \mathbf{N}$은 $\gcd(n,m)=1$이다. 그러면*

$$\begin{aligned} x &\equiv a \pmod m, \\ x &\equiv b \pmod n \end{aligned}$$

을 만족하는 정수해 $x \in \mathbf{Z}$가 존재하고 그 정수해 x는 법 mn에 관해서 유일하다.

증명 다음 방정식

$$a + tm \equiv b \pmod{n}$$

을 만족하는 정수해 t를 찾을 수 있다면 $x = a+tm$은 위의 두 합동식을 만족함은 쉽게 확인할 수 있다. 그런데 위 식은 양쪽에 a를 빼면 $tm \equiv b-a \pmod{n}$인데 기초정리 2.1.13와 $\gcd(n, m) = 1$이라는 정리의 가정을 이용하면 항상 해를 갖는다. 유일성을 보이기 위하여 x와 y가 정리의 연립합동식의 해라고 하자. 그러면 $z = x - y$는 $z \equiv 0 \pmod{m}$와 $z \equiv 0 \pmod{n}$를 만족하므로 $m \mid z$ and $n \mid z$이다. $\gcd(n, m) = 1$이므로 $nm \mid z$을 얻고 따라서 $x \equiv y \pmod{nm}$이 성립한다. □

알고리듬 2.2.3 (중국인의 나머지 정리)**.** 서로 소인 두 정수 m, n과 정수 a와 b를 줄 때 $x \equiv a \pmod{m}$, $x \equiv b \pmod{n}$를 동시에 만족하는 x를 구하는 알고리듬이다.

1. [Extended GCD] 아래의 Algorithm 2.3.7을 사용하여 $cm + dn = 1$를 만족하는 정수 c, d를 구한다.
2. [Answer] $x = a + (b - a)cm$를 출력하고 알고리듬을 끝낸다.

증명 $c \in \mathbf{Z}$이므로, $x \equiv a \pmod{m}$이고, 또 $cm + dn = 1$을 사용하면 $a + (b - a)cm \equiv a + (b - a) \equiv b \pmod{n}$를 얻는다. □

이제 질문 2.2.1에 대한 답을 할 수 있다. 먼저, 정리 2.2.2를 이용하여 다음 두 개의 합동식

$$x \equiv 2 \pmod{3},$$
$$x \equiv 3 \pmod{5}$$

의 해를 찾는다. $a = 2$, $b = 3$, $m = 3$, $n = 5$라 놓자. $x \equiv 2 \pmod{3}$의 해는 $x = 2 + 3t$이므로 1단계는 $t \cdot 3 \equiv 3 - 2 \pmod{5}$의 해 t를 찾는다. $t = 2$가 해이다. (a, b, m, n으로 표현하면 $t \cdot m = b - a \pmod{n}$) 그러면 $x = a + tm = 2 + 2 \cdot 3 = 8$이다. 또, $x' \equiv x \pmod{15}$를 만족하는 모든 x'는 위의 두 합동식의 해가 되므로, 다음 두 합동식

$$x \equiv 8 \pmod{15},$$
$$x \equiv 2 \pmod{7}$$

의 공통해를 구하면 처음에 주어진 세 식의 공통해를 구하는 것이 된다. (여기서는 $a = 8, b = 2, m = 15, n = 7$이므로) 한번 더 $t \cdot 15 \equiv 2 - 8 \pmod{7}$의 해 $t = 1$를 구하고

$$x = a + tm = 8 + 15 = 23$$

를 계산한다. 임의의 $x' \equiv x \pmod{3 \cdot 5 \cdot 7}$ 도 해가 되므로 다른 해들도 존재함을 확인하자. 예를 들어 $23 + 3 \cdot 5 \cdot 7 = 128$은 또 다른 해이다.

SAGE 예 2.2.4. Sage에서 `CRT(a,b,m,n)`는 $x \equiv a \pmod{m}$와 $x \equiv b \pmod{n}$의 공통해를 계산한다.

```
sage: CRT(2,3, 3, 5)
8
```

세 개 이상의 연립 합동식은 `CRT_list`를 이용한다. 이 절을 시작 문제 2.2.1는 다음 명령어로 얻을 수 있다.

```
sage: CRT_list([2,3,2], [3,5,7])
23
```

2.2.1 곱셈 함수

정의 2.1.18의 Euler φ-함수는 다음과 같다.

$$\varphi(n) = \#\{a : 1 \le a \le n \text{ and } \gcd(a,n) = 1\}.$$

보조정리 2.2.5. *$m, n \in \mathbf{N}$이고 $\gcd(m,n) = 1$이라고 가정하자. 그러면*

$$\psi(c) = (c \bmod m,\ c \bmod n)$$

로 정의한 함수

$$\psi : (\mathbf{Z}/mn\mathbf{Z})^* \to (\mathbf{Z}/m\mathbf{Z})^* \times (\mathbf{Z}/n\mathbf{Z})^* \tag{2.2.1}$$

는 일대일 대응이다.

증명 먼저 ψ가 일대일 함수임을 보이자. 만약 $\psi(c) = \psi(c')$이면, $m \mid c - c'$과 $n \mid c - c'$이 성립한다. 가정에 의하여 $\gcd(n,m) = 1$이므로, $nm \mid c - c'$가 성립한다. 따라서 $(\mathbf{Z}/mn\mathbf{Z})^*$의 원소로서 $c = c'$이다.

이제 ψ가 전사임을 보이자. 즉, $(\mathbf{Z}/m\mathbf{Z})^* \times (\mathbf{Z}/n\mathbf{Z})^*$의 모든 원소는 적당한 c가 있어 $\psi(c)$로 표현됨을 보이면 된다. $\gcd(a,m) = 1$이고 $\gcd(b,n) = 1$인 a와 b가 주어지면, 중국인의 나머지 정리 2.2.2로부터 $c \equiv a \pmod m$와 $c \equiv b \pmod n$를 만족하는 c가 존재한다. $1 \le c \le nm$이라고 가정할 수 있고, 또 $\gcd(a,m) = 1$, $\gcd(b,n) = 1$이므로, 반드시 $\gcd(c,nm) = 1$이어야만 한다. 따라서 $\psi(c) = (a,b)$이다. □

정의 2.2.6 (곱셈함수)**.** 함수 $f : \mathbf{N} \to \mathbf{C}$가 $\gcd(m,n) = 1$인 두 자연수 $m, n \in \mathbf{N}$에 대하여,

$$f(mn) = f(m) \cdot f(n).$$

를 만족하면 **곱셈함수(multiplicative function)**라고 한다.

기초정리 2.2.7 (곱셈함수 φ)**.** *오일러 함수 φ는 곱셈함수이다.*

증명 보조정리 2.2.5의 함수 ψ는 전단사 함수이므로 (2.2.1)의 왼쪽 집합이나 (2.2.1)의 오른쪽 집합이나 같은 크기이다. 따라서

$$\varphi(mn) = \varphi(m) \cdot \varphi(n)$$

이 성립한다. □

이 기초정리는, 적어도 우리가 n을 소인수분해할 수 있다면, $\varphi(n)$을 계산할 때 아주 유용하다. (n을 인수분해하는 것과 $\varphi(n)$의 값을 구하는 것과의 관계는 절 3.4.1을 참고한다.) 예를 들어

$$\varphi(12) = \varphi(2^2) \cdot \varphi(3) = 2 \cdot 2 = 4.$$

또 $n \geq 1$일 때,

$$\varphi(p^n) = p^n - \frac{p^n}{p} = p^n - p^{n-1} = p^{n-1}(p-1) \tag{2.2.2}$$

이 성립한다. 왜냐하면 p^n보다 작은 수 중에서 p와 서로 소가 아닌 수는 p의 배수뿐이기 때문이다. 그러므로, 예를 든다면,

$$\varphi(389 \cdot 11^2) = 388 \cdot (11^2 - 11) = 388 \cdot 110 = 42680.$$

2.3 역원과 아주 큰 거듭제곱의 빠른 계산법

이 절에서는 합동식 $ax \equiv 1 \pmod{n}$이 해를 가지는 것을 아는 경우에 해를 구하는 방법과 $a^m \pmod{n}$의 효율적인 계산 방법을 살펴본다. 또 $a^m \pmod{n}$을 빨리 계산할 수 있는 우리의 능력에 의존하는 간단한 소수 판정법에 대해 논한다. 이 세 가지 알고리듬은 모두 단원 3의 암호 알고리듬의 기본이 되는 중요한 알고리듬들이다.

2.3.1 $ax \equiv 1 \pmod{n}$의 풀이법

$a, n \in \mathbf{N}$은 $\gcd(a,n) = 1$인 정수들이다. 그러면 기초정리 2.1.13에 의하여 일차 합동식 $ax \equiv 1 \pmod{n}$은 유일한 해를 가진다. 이 해를 구하는 방법은?

기초정리 2.3.1 (확장된 유클리드 표현)**.** *$a, b \in \mathbf{Z}$이고 $g = \gcd(a,b)$라고 하자. 그러면*

$$ax + by = g.$$

를 만족하는 $x, y \in \mathbf{Z}$가 존재한다.

참조 2.3.2. 만약 $e = cg$이면 $cax + cby = cg = e$이므로, $e = (cx)a + (cy)b$와 같이 e 역시 a와 b로 표현된다.

증명 [기초정리 2.3.1의 증명] $g = \gcd(a,b)$이면 $\gcd(a/g, b/g) = 1$이므로, 기초정리 2.1.15에 의하여 , 합동식

$$\frac{a}{g} \cdot x \equiv 1 \left(\mathrm{mod}\ \frac{b}{g} \right) \tag{2.3.1}$$

은 해 $x \in \mathbf{Z}$를 갖는다. 식 (2.3.1)에 g를 곱하면 $ax \equiv g \pmod{b}$를 만족하므로 $b \cdot (-y) = ax - g$를 만족하는 y가 존재한다. 따라서 $ax + by = g$를 만족하는 x, y가 존재한다. □

기초정리 2.3.1의 증명은, 법 n에서 일차 합동식을 푸는 알고리듬을 안다고 가정하면, $g = \gcd(a,b)$일 때 $ax+by=g$를 만족하는 정수 x, y를 구체적으로 찾을 수 있는 방법을 제시한다. 아직은 그런 알고리듬을 모르므로 x와 y를 구체적으로 찾을 수 있는 방법에 대해 이제부터 알아보고자 한다. 이 알고리듬은 사실 법 n 일차합동식도 풀 수 있게 한다. $\gcd(a,n)=1$일 때 $ax \equiv 1 \pmod{n}$을 풀려면, $ax+ny=1$을 만족하는 x와 y를 구하는 알고리듬 2.3.7를 사용한다.

예 2.3.3. $a=5$, $b=7$일 때 $\gcd(5,7)$를 계산하는 알고리듬 1.1.13의 각 단계는 다음과 같다. 어떤 수에는 밑줄이 쳐 있는데 x 와 y를 구하기 위하여 아래 식에서부터 대입할 때 눈에 띄게 하기 위해서이다.

$$\underline{7} = 1\cdot\underline{5} + \underline{2}\text{이므로} \qquad \underline{2} = \underline{7} - \underline{5}$$
$$\underline{5} = 2\cdot\underline{2} + \underline{1}\text{이므로} \qquad \underline{1} = \underline{5} - 2\cdot\underline{2} = \underline{5} - 2(\underline{7}-\underline{5}) = 3\cdot\underline{5} - 2\cdot\underline{7}$$

오른쪽 식은 각 단계에서의 나머지를 a와 b의 일차결합으로 나타내기 위한 계산이다. 마지막 단계에서 $\gcd(a,b)$를 우리가 원하는 a와 b의 일차결합으로 나타낸다.

예 2.3.4. 앞의 예보다는 약간 복잡한 경우를 계산해보자. $a=130$, $b=61$으로 놓으면

$$\begin{aligned} \underline{130} &= 2\cdot\underline{61} + \underline{8} & \underline{8} &= \underline{130} - 2\cdot\underline{61} \\ \underline{61} &= 7\cdot\underline{8} + \underline{5} & \underline{5} &= -7\cdot\underline{130} + 15\cdot\underline{61} \\ \underline{8} &= 1\cdot\underline{5} + \underline{3} & \underline{3} &= 8\cdot\underline{130} - 17\cdot\underline{61} \\ \underline{5} &= 1\cdot\underline{3} + \underline{2} & \underline{2} &= -15\cdot\underline{130} + 32\cdot\underline{61} \\ \underline{3} &= 1\cdot\underline{2} + \underline{1} & \underline{1} &= 23\cdot\underline{130} - 49\cdot\underline{61} \end{aligned}$$

을 얻는다. 따라서 $x=23$, $y=-49$가 $130x+61y=1$의 해이다.

예 2.3.5. 이 예는 바로 앞의 예 2.3.4에서 오른쪽의 식을 벡터로 간단하게 정리 한 것이다.

$$\begin{aligned} \underline{130} &= 2\cdot\underline{61} + \underline{8} & \underline{8} &= (1,-2) \\ \underline{61} &= 7\cdot\underline{8} + \underline{5} & \underline{5} &= (-7,15) = (0,1) - 7(1,-2) \\ \underline{8} &= 1\cdot\underline{5} + \underline{3} & \underline{3} &= (8,-17) = (1,-2) - (-7,15) \\ \underline{5} &= 1\cdot\underline{3} + \underline{2} & \underline{2} &= (-15,32) = (-7,15) - (8,-17) \\ \underline{3} &= 1\cdot\underline{2} + \underline{1} & \underline{1} &= (23,-49) = (8,-17) - (-15,32) \end{aligned}$$

각 단계에서 오른쪽의 벡터는 두 단계 앞의 벡터에서 한 단계 앞의 벡터의 적당한 상수배를 뺀 것인데 이 상수는 각 단계에서의 몫이다. 각 단계에서 오른쪽의 벡터는 두 단계 앞의 벡터에서 한 단계 앞의 벡터에 이 줄에서 얻은 몫만큼의 상수배를 하여 뺀 것이다.

SAGE 예 2.3.6. `xgcd(a,b)` 명령어는 a와 b의 최대공약수 g와 $ax+by=g$를 만족하는 $x,\ y$를 계산한다.

```
sage: xgcd(5,7)
(1, 3, -2)
sage: xgcd(130,61)
(1, 23, -49)
```

알고리듬 2.3.7 (확장된 유클리드 알고리듬)**.** a와 b가 정수이고 $g = \gcd(a, b)$라고 하자. 이 알고리듬은 $ax + by = g$인 g, x, 그리고 y를 계산한다. 항상 $a > b \geq 0$라고 가정할 수 있으므로 이 경우에서 이 알고리듬의 각 단계를 설명한다.

1. [시작] Set $x = 1$, $y = 0$, $r = 0$, $s = 1$로 놓는다.
2. [끝?] If $b = 0$이면 $g = a$이라 놓고 마친다.
3. [몫과 나머지] 알고리듬 1.1.12을 이용하여 $0 \leq c < b$가 되도록 $a = qb+c$의 형태로 쓴다.
4. [바꾸기] $(a, b, r, s, x, y) = (b, c, x-qr, y-qs, r, s)$라 놓고 단계 2로 간다. (이 옮기는 단계는 예에 멋지게 예시되어 있다.)

증명 이 알고리듬은, 우리가 더 많은 변수 x, y, r, s의 값도 계속 추적한다는 것을 제외하고는, 알고리듬 1.1.13와 같으므로 $d = \gcd(a, b)$를 주고 끝난다. 이 알고리듬의 귀납적 증명은 생략한다. 관심 있는 독자들은 [27, §1.2.1]를 참고하기 바란다. □

알고리듬 2.3.8 (법 n 역원)**.** a와 n은 정수이고 $\gcd(a, n) = 1$이라고 가정하자. 그러면 이 알고리듬은 $ax \equiv 1 \pmod{n}$을 만족하는 x를 찾는다.

1. [확장된 GCD 계산] 알고리듬 2.3.7을 이용하여 $ax+ny = \gcd(a, n) = 1$를 만족하는 정수 x, y를 계산한다.
2. [마침] x를 출력한다.

증명 $ax+ny = 1$를 법 n으로 축약하면 x가 $ax \equiv 1 \pmod{n}$을 만족한다. □

예 2.3.9. $17x \equiv 1 \pmod{61}$를 풀어라. 먼저 알고리듬 2.3.7을 사용하여 $17x+61y = 1$를 만족하는 x, y를 구한다:

$$\begin{aligned} \underline{61} &= 3 \cdot \underline{17} + \underline{10} & \qquad \underline{10} &= \underline{61} - 3 \cdot \underline{17} \\ \underline{17} &= 1 \cdot \underline{10} + \underline{7} & \qquad \underline{7} &= -\underline{61} + 4 \cdot \underline{17} \\ \underline{10} &= 1 \cdot \underline{7} + \underline{3} & \qquad \underline{3} &= 2 \cdot \underline{61} - 7 \cdot \underline{17} \\ \underline{3} &= 2 \cdot \underline{3} + \underline{1} & \qquad \underline{1} &= -5 \cdot \underline{61} + 18 \cdot \underline{17}. \end{aligned}$$

따라서 $17 \cdot 18 + 61 \cdot (-5) = 1$이므로 $x = 18$이 $17x \equiv 1 \pmod{61}$의 해이다.

SAGE 예 2.3.10. Sage에서는 위의 알고리듬을 구현하여 법 n 역원을 찾는다. 예를 들어

```
sage: a = Mod(17, 61)
sage: a^(-1)
18
```

2.3.2 $a^m \pmod n$*의 계산*

a 와 n은 정수이고 m은 음수가 아닌 정수이다. 이 절에서는 $a^m \pmod n$을 계산하는 효율적인 알고리듬을 설명한다. 단원 3에서 암호론에 응용할 때는 m은 몇 백 자릿수이다.

$a^m \pmod n$을 계산하는 단순한 접근은 a를 계속 곱하여 법 n에 관한 나머지를 계산하는 것을 계속하여 $a^m = a \cdot a \cdots a \pmod n$을 계산하는 것이다. 각 연산마다 법 n에 관한 나머지만을 남기므로 이 계산 중에 너무 큰 수는 나타나지 않는다. 그럼에도 불구하고 이 방법은 m번의 곱셈을 해야하므로 m이 큰 수일 때는 아주 비효율적이다.

m을 이진법으로 표현하여 a^m을 여러 a^{2^i}꼴의 곱으로 표현하여 훨씬 효율적인 알고리듬을 얻을 수 있다. a^{2^i}는 a부터 시작하여 계속 제곱함으로써 얻을 수 있다. 이 똑똑한 알고리듬은 간단하지는 않지만 전자의 연산수가 $m-1$인데 비하여 후자의 연산수는 m의 이진법 표현의 자릿수에 비례하여 커지므로 훨씬 효율적이다.

알고리듬 2.3.11 (이진법으로 수 쓰기)**.** 이 알고리듬은 음수가 아닌 정수 m을 이진법으로 표현한다. 따라서 이 알고리듬은 $\varepsilon_i \in \{0,1\}$이면서 $m = \sum_{i=0}^{r} \varepsilon_i 2^i$이 되는 $\varepsilon_i \in \{0,1\}$를 찾아낸다.

1. [시작] $i = 0$으로 놓는다.
2. [끝?] $m = 0$이면 마친다.
3. [자릿수 값 계산] m이 홀수이면 $\varepsilon_i = 1$, 짝수이면 $\varepsilon_i = 0$로 놓는다. i를 올린다.
4. [2로 나누기] $m = \left\lfloor \frac{m}{2} \right\rfloor (\leq m/2$인 가장 큰 정수)로 놓고 단계 2로 간다.

SAGE 예 2.3.12. Sage를 이용하여 주어진 수를 이진법으로 표현하는 명령어는 `str`이다.

```
sage: 100.str(2)
'1100100'
```

위의 식은 올바른 이진법 표현임을 확인하도록.

```
sage: 0*2^0 + 0*2^1 + 1*2^2 + 0*2^3 + 0*2^4 + 1*2^5 + 1*2^6
100
```

알고리듬 2.3.13 (거듭제곱계산)**.** a와 n은 정수, m은 음수가 아닌 정수일 때, 이 알로리듬은 법 n에서 a^m을 계산한다.

1. [이진수로 표현] 알고리듬 2.3.11을 이용하여 m을 다음과 같이 $a^m = \prod_{\varepsilon_i=1} a^{2^i} \pmod n$ 이진법으로 표현.
2. [거듭제곱 계산] $r+1$이 m의 이진법 자릿수라면, a, a^2, $a^{2^2} = (a^2)^2$, $a^{2^3} = (a^{2^2})^2$, $\ldots$, a^{2^r}을 계산한다.
3. [거듭제곱곱하기] $\varepsilon_i = 1$인 a^{2^i}들을 모두 곱한다. 모든 계산은 항상 법 n 계산이다.

예 2.3.14. $7^{91} \pmod{100}$를 계산함으로써 7^{91}의 마지막 두 자릿수를 알 수 있다. $\gcd(7, 100) = 1$이므로, 정리 2.1.20로부터 $7^{\varphi(100)} \equiv 1 \pmod{100}$이다. φ는 곱셈함수이므로,

$$\varphi(100) = \varphi(2^2 \cdot 5^2) = (2^2 - 2) \cdot (5^2 - 5) = 40.$$

따라서 $7^{40} \equiv 1 \pmod{100}$이고

$$7^{91} \equiv 7^{40+40+11} \equiv 7^{11} \pmod{100}.$$

이제 위의 알고리듬을 이용하여 $7^{11} \pmod{100}$을 다음과 같이 계산한다. 먼저, 11을 2로 계속 나눈다.

$$\begin{aligned} 11 &= 5 \cdot 2 + 1 \\ 5 &= 2 \cdot 2 + 1 \\ 2 &= 1 \cdot 2 + 0 \\ 1 &= 0 \cdot 2 + 1 \end{aligned}$$

따라서, $(11)_2 = 1011$, 이며, 검산하면

$$11 = 1 \cdot 8 + 1 \cdot 2 + 1.$$

다음, a, a^2, a^4, a^8을 계산하고 $a^8 \cdot a^2 \cdot a$ 를 출력한다. 그러면

$$\begin{aligned} a &= 7 \\ a^2 &\equiv 49 \\ a^4 &\equiv 49^2 \equiv 1 \\ a^8 &\equiv 1^2 \equiv 1 \end{aligned}$$

(49를 제곱하는 가장 쉬운 계산은 법 4와 25로 각각 계산한 다음 중국인의 나머지 정리를 이용하는 것임을 참고하길.) 그러면 우리가 원하는 답은 다음과 같다.

$$7^{91} \equiv 7^{11} \equiv a^8 \cdot a^2 \cdot a \equiv 1 \cdot 49 \cdot 7 \equiv 43 \pmod{100}.$$

SAGE 예 2.3.15. Sage에서는 위의 알고리듬을 실행시켜 거듭제곱을 구한다. 예를 들어,

```
sage: Mod(7,100)^91
43
```

물론 Sage에서 7^{91} 을 바로 구할 수도 있다. 누구도 이 값을 손으로 직접 계산하려고 하지는 않겠지만...

```
sage: 7^91
80153343160247310515380886994816022539378033762994852
00750196460484168019074 3
```

2.4 소수 테스트

정리 2.4.1. *1보다 큰 정수 $p > 1$가 소수이기 위한 필요충분조건은 $a \not\equiv 0 \pmod{p}$인 모든 정수 a가*

$$a^{p-1} \equiv 1 \pmod{p}.$$

를 만족하는 것이다.

증명 p가 소수이면 기초정리 2.1.22으로부터 필요조건이 따라온다. p가 합성수이면 p의 약수 a가 존재하고 $2 \leq a < p$이다. 만약 $a^{p-1} \equiv 1 \pmod{p}$이라면 $p \mid a^{p-1} - 1$을 만족한다. $a \mid p$이므로 $a \mid a^{p-1} - 1$가 성립하고, 따라서 $(ak = a^{p-1} - 1$인 정수 k가 존재한다. $a^{p-1} - ak = 1$이므로 $a(a^{p-2} - k) = 1$을 얻고 이 식은) $a \mid 1$을 유도하므로 $2 \leq a < p$인 사실에 모순이다. □

$n \in \mathbf{N}$이라고 하자. 그러면 정리 2.4.1와 알고리듬 2.3.13을 이용하여 n이 소수가 아닌 것을 빠르게 증명하거나, 혹은 (n이 소수라고 빠르게 증명할 수는 없지만) 소수일 것 같다고 확신할 수는 있다. 예를 들어, $2^{n-1} \not\equiv 1 \pmod{n}$이면 n이 소수가 아님이 증명된다. 반면에 몇 개의 a에 대하여 $a^{n-1} \equiv 1 \pmod{n}$이면 n이 소수일 수도 있다고 생각할 수 있을 뿐이다. 이런 수를 **유사소수(pseudoprime)**라고 부른다.

합성수 중에서 $\gcd(a, n) = 1$을 만족하는 *모든* a에 대하여 $a^{n-1} \equiv 1 \pmod{n}$을 만족하는 이런 놀라운 성질을 갖는 정수 n이 존재하는데 이런 수를 **Carmichael 수**라고 부른다. 첫 번째 Carmichael 수는 561이며, 이런 수들이 무한히 많음이 알려져 있다 [2].

예 2.4.2. $p = 323$는 소수인가? $2^{322} \pmod{323}$를 계산한다. 다음은 이 값을 위에서 설명한대로 알고리듬2.3.13으로 계산하기 위한 표이다.

i	m	ε_i	$2^{2^i} \bmod 323$
0	322	0	2
1	161	1	4
2	80	0	16
3	40	0	256
4	20	0	290
5	10	0	120
6	5	1	188
7	2	0	137
8	1	1	35

따라서

$$2^{322} \equiv 4 \cdot 188 \cdot 35 \equiv 157 \pmod{323}$$

이므로 323은 소수가 아니다. 물론 이 계산은 323의 소인수 분해에 대한 어떤 정보도 주지 못한다. 사실 $323 = 17 \cdot 19$임을 우리는 보일 수 있다.

SAGE 예 2.4.3. 큰 수의 약수를 찾지는 못하면서 합성수임은 쉽게 보일 수 있다. 예를 들어

$$n = 954680934860934509834095834098509348509384590 83$$

은 $2^{n-1} \not\equiv 1 \pmod{n}$이므로 n은 합성수이다.

```
sage: n = 954680934860934509834095834098509348509384590 83
sage: Mod(2,n)^(n-1)
341734441392655538708302663785984070692486872 41
```

n을 인수분해하는데는 훨씬 더 긴 시간이 소요된다.

```
sage: factor(n)                  # takes up to a few seconds.
1610302526747 * 592858123864154884463971917910230 89
```

또 다른 실용적인 소수 판정법으로 Miller-Rabin test가 있는데, 이 판정법을 어떤 수 n에 적용하면 정확히 소수가 아니라고 알려주거나, 혹은 시행 횟수에 의존하는 확률로 소수일 것이다고 알려준다. 만약 자연수 n에 Miller-Rabin 판정을 m번 시행하여 아마도 소수일 수도 있다고 나타난다면 소수일 확률을 n과 m에 따라 정확히 표현할 수 있다.

우리는 Miller-Rabin 알고리듬을 정확히 기술은 하겠지만 이 알고리듬에서 확률에 관한 내용은 증명하지 않는다.

알고리듬 2.4.4 (Miller-Rabin의 소수 판정)**.** 5보다 크거나 같은 정수 $n \geq 5$에 대하여 이 알고리듬은 true 혹은 false를 출력한다. true를 출력하면, n은 "아마도 소수(probably prime)"이고, false를 출력하면, n는 정확히 합성수이다.

1. [2의 거듭제곱 분할] $n-1 = 2^k \cdot m$을 2의 거듭제곱과 홀수 m의 곱으로 표현하는 유일한 k와 m을 계산한다.
2. [밑수 선택] $1 < a < n$인 a를 무작위로 선택한다.
3. [홀수 거듭제곱] $b = a^m \pmod{n}$로 놓는다. $b \equiv \pm 1 \pmod{n}$이면 true를 출력하고 끝낸다.
4. [짝수 거듭제곱들] $1 \leq r \leq k-1$인 어떤 r에서 $b^{2^r} \equiv -1 \pmod{n}$이면 true를 출력하고 끝낸다. 그렇지 않으면 false를 출력한다.

n에 대한 Miller-Rabin이 true를 출력하면, 한 번 더 Miller-Rabin 테스트를 하고 또 true를 출력하면 n을 소수라고 잘못 판정하는 확률은 감소한다.

증명 이 알고리듬이 참임을 보인다. 그러나 어떻게 합성수를 소수라고 주장할 수 있는지는 전혀 증명하지 않는다. 우리는 이 알고리듬이 n을 합성수라고 말하면 n은 합성수임을 보여야만 한다. 따라서 n이 소수인데 이 알고리듬이 n을 합성수라고 판정했다고 가정하자. 그러면 $a^m \not\equiv \pm 1 \pmod{n}$이고, 또 모든 $1 \leq r \leq k-1$인 r에 대해서 $a^{2^r m} \not\equiv -1 \pmod{n}$을 만족한다. n이 소수이고 $2^{k-1}m = (n-1)/2$이므로, 기초정리 4.2.1로부터 $a^{2^{k-1}m} \equiv \pm 1 \pmod{n}$가

성립하여, 우리의 가정으로부터, $a^{2^{k-1}m} \equiv 1 \pmod{n}$을 얻는다. 그러나 그러면 $(a^{2^{k-2}m})^2 \equiv 1 \pmod{n}$이므로, 기초정리 2.5.3에 의하여, $a^{2^{k-2}m} \equiv \pm 1 \pmod{n}$이 성립한다 (기초정리 2.5.3는 이 알고리듬의 증명 바로 뒤에 나오긴 하지만 지금 증명과는 독립적으로 증명이 되는 것이므로 이용 가능함). 우리의 가정을 한 번 더 적용하면 $a^{2^{k-2}m} \equiv 1 \pmod{n}$이다. 이 주장을 귀납적으로 되풀이하면 $a^m \equiv \pm 1 \pmod{n}$을 얻는데 이는 a의 조건에 모순이다. □

최근까지 정수의 자릿수에 관한 다항식 시간안에 하는 임의의 정수의 소수 판정 알고리듬이 존재하는 지는 알려져 있지 않았다. Agrawal, Kayal, and Saxena이 최근에 다항식 시간 소수 판정법을 처음으로 발견하였다 ([3]). 이 책에서 다룰 암호 알고리듬에 대한 응용으로는 Miller-Rabin 혹은 유사소수 판정법으로도 충분하므로 이들의 알고리듬을 여기서는 더 기술하지는 않는다. 이 알고리듬은 [47, Ch. 21]에 자세히 설명되어 있으니 참고하기 바란다.

SAGE 예 2.4.5. Sage에서 `is_prime` 함수는 주어진 정수가 소수인지 아닌지를 결정한다.

```
sage: n = 954680934860934509834095834098509348509384590 83
sage: is_prime(n)
False
```

`is_prime` 함수를 이용하여 처음 몇 개의 메르센 소수표를 만들어 보자. (절 1.2.3 참조).

```
sage: for p in primes(100):
...   if is_prime(2^p - 1):
...       print p, 2^p - 1
2 3
3 7
5 31
7 127
13 8191
17 131071
19 524287
31 2147483647
61 2305843009213693951
89 618970019642690137449562111
```

Mersenne 수에 대한 특화된 소수 판정법이 있다. 이 판정법은 Lucas-Lehmer 판정법인데 $2^p - 1$이 소수인지 아닌지를 거의 바르게 판정하는 아주 간단한 판정법이다. 몇 줄의 명령어를 실행하고 Lucas-Lehmer 판정법을 사용하여 두 메르센 수의 소수 여부를 판정한다.

```
sage: def is_prime_lucas_lehmer(p):
...    s = Mod(4, 2^p - 1)
...    for i in range(3, p+1):
...        s = s^2 - 2
```

```
...    return s == 0
sage: # Check primality of 2^9941 - 1
sage: is_prime_lucas_lehmer(9941)
True
sage: # Check primality of 2^next_prime(1000)-1
sage: is_prime_lucas_lehmer(next_prime(1000))
False
```

메르센 소수에 대하여 더 알고 싶으면, `http://www.mersenne.org/`에서 the Great Internet Mersenne Prime Search (GIMPS) 과제를 참고하기 바란다.

2.5 $(\mathbf{Z}/p\mathbf{Z})^*$의 구조

이 절은 p가 소수일 때 법 p 곱셈으로 군이 되는 집합 $(\mathbf{Z}/p\mathbf{Z})^*$의 구조를 다룬다. 핵심 결과는, 군의 용어를 빌리면, 이 군은 **순환군(cyclic group)** [2]이라는 것이다. 이 결과는 뒤에 단원 4에서 이차상호법칙을 증명할 때 사용한다.

정의 2.5.1 (원시근)**.** 법 n의 **원시근(primitive root)**은 위수가 $\varphi(n)$인 $(\mathbf{Z}/n\mathbf{Z})^*$의 원소이다.

모든 소수 p는 원시근을 가짐을 보일 것이다. $(\mathbf{Z}/p\mathbf{Z})^*$은 $p-1$개의 원소를 가짐으로, 원시근의 존재는 $(\mathbf{Z}/p\mathbf{Z})^*$이 순환군임을 유도한다.

만약 n이 홀수인 소수의 거듭제곱이면, 법 n은 원시근을 갖는다(Exercise 2.28 참조). 그러나 2^3은 원시근을 갖지 않고, 따라서 $n \geq 3$인 2^n도 원시근을 갖지 않는다 (Exercise 2.27 참조).

절 2.5.1은 $(\mathbf{Z}/p\mathbf{Z})^*$이 순환군임을 보이는데 아주 결정적 역할을 한다. 여기서 $p-1$의 모든 약수 d에 대하여 $(\mathbf{Z}/p\mathbf{Z})^*$의 원소 중에서 위수가 d의 약수인 원소가 정확히 d 개임을 보인다. 그런 후 절 2.5.2에 있는 이 결과를 이용하여, 소수의 거듭제곱인 q^r이 $p-1$을 나누는 가장 큰 거듭제곱일 때, 위수가 q^r인 $(\mathbf{Z}/p\mathbf{Z})^*$의 원소를 만들어낸다. 그리고 이 수들을 모두 곱하여 위수가 $p-1$인 $(\mathbf{Z}/p\mathbf{Z})^*$의 원소를 얻는다.

SAGE 예 2.5.2. Sage 코드 `primitive_root`를 이용하면 법 n의 가장 작은 원시근을 계산할 수 있다. 다음은 $p < 20$인 법 p의 원시근이다.

```
sage: for p in primes(20):
...    print p, primitive_root(p)
2 1
3 2
5 2
7 3
11 2
```

[2] $(\mathbf{Z}/p\mathbf{Z})^*$의 모든 원소가 g의 거듭제곱으로 표현되는 $g \in (\mathbf{Z}/p\mathbf{Z})^*$가 존재할 때 순환군이라고 하고 $(\mathbf{Z}/p\mathbf{Z})^* = (g)$로 표현. 이 g가 원시근이다.

```
13 2
17 3
19 2
```

2.5.1 법 p에 관한 다항식

다항식 x^2-1은 $\mathbf{Z}/8\mathbf{Z}$에서 1, 3, 5, 7 네 개의 해를 갖는다. 대조적으로 다음 기초정리는 차수가 d인 다항식은 체 안에서는 많아야 d 개의 해만을 가질수 있음을 보여준다. [3]

기초정리 2.5.3 (다항식의 해의 개수)**.** *k는 체이고 $f \in k[x]$는 영이 아닌 다항식이다. 그러면 $f(\alpha)=0$을 만족하는 k의 원소 $\alpha \in k$ 많아야 $\deg(f)$ 개 존재한다.*

증명 $\deg(f)$에 관한 귀납법으로 증명하자. $\deg(f) \leq 1$인 경우는 자명하다. 이제 $f = a_n x^n + \cdots a_1 x + a_0$로 쓰자. 만약 $f(\alpha)=0$이면,

$$\begin{aligned} f(x) &= f(x) - f(\alpha) \\ &= a_n(x^n - \alpha^n) + \cdots + a_1(x-\alpha) + a_0(1-1) \\ &= (x-\alpha)(a_n(x^{n-1} + \cdots + \alpha^{n-1}) + \cdots + a_2(x+\alpha) + a_1) \\ &= (x-\alpha)g(x) \end{aligned}$$

로 쓸 수 있다. 이 때 $g(x) \in k[x]$이다. 다음, $\beta \neq \alpha$이면서 $f(\beta)=0$이라고 하자. 그러면, $(\beta-\alpha)g(\beta)=0$인데, $\beta-\alpha \neq 0$이고 k는 체이므로, $g(\beta)=0$ 이 된다. 귀납적 가정에 의하여, g는 많아야 $n-1$ 개의 해를 가지므로, 많아야 $n-1$개의 β가 존재할 수 있다. 따라서 f는 많아야 n 개의 해를 갖는다. □

SAGE 예 2.5.4. Sage를 이용하여 $\mathbf{Z}/13\mathbf{Z}$에 계수를 갖는 다항식의 해를 구하자.

```
sage: R.<x> = PolynomialRing(Integers(13))
sage: f = x^15 + 1
sage: f.roots()
[(12, 1), (10, 1), (4, 1)]
sage: f(12)
0
```

해를 출력할 때 각 해에 중복도를 함께 출력한다. 위의 예에서는 모든 해의 중복도가 1이다.

기초정리 2.5.5. *p는 소수이고 d는 $p-1$의 약수이다. 그러면 $f = x^d - 1 \in (\mathbf{Z}/p\mathbf{Z})[x]$는 $\mathbf{Z}/p\mathbf{Z}$에서 정확히 d 개의 해를 갖는다.*

---

[3] 예를 들어 p가 소수이면, 또 오직 그 경우에만 $\mathbf{Z}/p\mathbf{Z}$는 체임

증명 $e = (p-1)/d$라 놓자. 그러면

$$\begin{aligned} x^{p-1} - 1 &= (x^d)^e - 1 \\ &= (x^d - 1)((x^d)^{e-1} + (x^d)^{e-2} + \cdots + 1) \\ &= (x^d - 1)g(x) \end{aligned}$$

이다. 이 때 $g \in (\mathbf{Z}/p\mathbf{Z})[x]$이고 $\deg(g) = de - d = p - 1 - d$이다. 정리 2.1.20로부터, $x^{p-1} - 1$은, 모든 $\mathbf{Z}/p\mathbf{Z}$의 원소가 해이므로, $\mathbf{Z}/p\mathbf{Z}$에서 정확히 $p-1$개의 해를 갖는다. 기초정리 2.5.3에 의하여, g는 *많아야* $p-1-d$ 개의 해를, $x^d - 1$은 많아야 d 개의 해를 갖는다. $(x^d - 1)g(x)$의 해는 $x^d - 1$이나 $g(x)$의 해야만 하고, $x^{p-1} - 1$은 정확히 $p-1$ 개의 해를 가지므로, g는 정확히 $p-1-d$ 개의 해를, 또 $x^d - 1$도 정확히 d 개의 해를 가져야만 한다. □

SAGE 예 2.5.6. Sage를 이용하여 기초정리의 성질을 예시해보자.

```
sage: R.<x> = PolynomialRing(Integers(13))
sage: f = x^6 + 1
sage: f.roots()
[(11, 1), (8, 1), (7, 1), (6, 1), (5, 1), (2, 1)]
```

이제 잠시 멈추어 소수 p를 합성수 n으로 바꾸면 기초정리 2.5.5는 더 이상 참이 아니라는 사실을 확인하자. 이유는 두 수의 곱이 법 n에서 0이라 하더라도 각각이 0이 될 필요가 없기 때문이다. 예를 들어 $f = x^2 - 1 = (x-1)(x+1) \in \mathbf{Z}/15\mathbf{Z}[x]$는 4 개의 해 1, 4, 11, 14를 갖는다.

2.5.2 원시근의 존재

절 2.1.2로부터 유한군의 원소 x의 **위수(order)**는 $x^m = 1$을 만족하는 가장 작은 양의 정수였다. 이 절에서는, 절 2.5.1를 이용하여 $p-1$의 소수 약수 d에 대하여 위수가 d인 $(\mathbf{Z}/p\mathbf{Z})^*$의 원소를 찾아 그들을 모두 곱하여 위수가 $p-1$인 원소를 발견하여, $(\mathbf{Z}/p\mathbf{Z})^*$가 순환군임을 보인다.

다음 보조정리를 이용하여 위수가 $p-1$의 약수인 원소들을 조립하여 위수가 $p-1$인 원소를 만들려고 한다.

보조정리 2.5.7. *$a, b \in (\mathbf{Z}/n\mathbf{Z})^*$의 위수가 각각 r과 s이고 $\gcd(r,s) = 1$이라고 하자. 그러면 ab의 위수는 rs이다.*

증명 이것은 임의의 군에서 서로 교환이 가능한 원소들에 대해 성립하는 일반적인 사실이다. 따라서 우리의 증명도 $ab = ba$인 사실만을 이용하고, $(\mathbf{Z}/n\mathbf{Z})^*$에만 성립하는 어떤 성질도 이용하지 않는다.

$$(ab)^{rs} = a^{rs}b^{rs} = 1$$

이므로, ab의 위수는 rs의 약수이다. 이 약수를 $r_1 s_1$로 쓰자. 이 때 물론 $r_1 \mid r$, $s_1 \mid s$이다.

$$a^{r_1 s_1} b^{r_1 s_1} = (ab)^{r_1 s_1} = 1$$

에 $r_2 = r/r_1$ 거듭제곱을 취한다.

$$a^{r_1 r_2 s_1} b^{r_1 r_2 s_1} = 1.$$

$a^{r_1 r_2 s_1} = (a^{r_1 r_2})^{s_1} = 1$이므로,

$$b^{r_1 r_2 s_1} = 1$$

이 성립한다. 따라서 $s \mid r_1 r_2 s_1$. 그런데 $\gcd(s, r_1 r_2) = \gcd(s, r) = 1$이므로, $s = s_1$이어야만 한다. 같은 방법으로 $r = r_1$을 얻을 수 있으므로, ab의 위수는 rs이다. □

정리 2.5.8 (원시근)**.** *p가 소수이면 법 p의 원시근이 존재한다. 특히, $(\mathbf{Z}/p\mathbf{Z})^*$는 순환군이다.*

증명 $p = 2$이면 1이 원시근이므로 이 정리는 성립한다. 따라서 $p > 2$라고 가정한다. $p - 1$를 서로 다른 소수들의 멱$q_i^{n_i}$의 곱으로 표현하면

$$p - 1 = q_1^{n_1} q_2^{n_2} \cdots q_r^{n_r}.$$

기초정리 2.5.5에 의하여, 다항식 $x^{q_i^{n_i}} - 1$은 정확히 $q_i^{n_i}$개의 해를 갖고, 다항식 $x^{q_i^{n_i-1}} - 1$은 정확히 $q_i^{n_i-1}$ 개의 해를 갖는다. 따라서 $\mathbf{Z}/p\mathbf{Z}$의 원소 중 위수가 $q_i^{n_i}$인 원소들의 개수는 $q_i^{n_i} - q_i^{n_i-1} = q_i^{n_i-1}(q_i - 1)$ 개이다. $a \in \mathbf{Z}/p\mathbf{Z}$가 그런 원소라면 $a^{q_i^{n_i}} = 1$이지만 $a^{q_i^{n_i-1}} \neq 1$을 만족한다. 따라서 $i = 1, \ldots, r$에 대하여, 위수가 $q_i^{n_i}$인 원소 a_i를 선택할 수 있다. 이제 보조정리 2.5.7를 계속 적용하면

$$a = a_1 a_2 \cdots a_r$$

는 위수가 $q_1^{n_1} \cdots q_r^{n_r} = p - 1$ 가 된다. 따라서 a는 법 p의 원시근이다. □

예 2.5.9. $p = 13$을 가지고 정리 2.5.8의 증명을 설명한다.

$$p - 1 = 12 = 2^2 \cdot 3.$$

다항식 $x^4 - 1$의 해는 $\{1, 5, 8, 12\}$이고, $x^2 - 1$의 해는 $\{1, 12\}$이므로 $a_1 = 5$로 잡자. 다항식 $x^3 - 1$의 해는 $\{1, 3, 9\}$이므로 $a_2 = 3$으로 잡을 수 있다. 그러면 $a = 5 \cdot 3 = 15 \equiv 2$가 원시근이다. 확인하기 위하여, $2 \pmod{13}$의 거듭제곱들을 다 계산해보면 다음과 같다.

$$2,\ 4,\ 8,\ 3,\ 6,\ 12,\ 11,\ 9,\ 5,\ 10,\ 7,\ 1.$$

예 2.5.10. 정리 2.5.8는 p가 4보다 큰 2의 거듭제곱이라면 성립하지 않는다. 예를 들어, $(\mathbf{Z}/8\mathbf{Z})^*$의 1이 아닌 모든 원소의 위수는 2이지만, $\varphi(8) = 4$이다.

정리 2.5.11 (법 p^n의 원시근)**.** *홀수 소수의 거듭제곱 p^n은 원시근을 갖는다.*

증명은 Exercise 2.28이다.

기초정리 2.5.12 (원시근의 개수)**.** *법 n이 원시근을 가지면, 법 n은 정확히 $\varphi(\varphi(n))$ 개의 원시근을 갖는다.*

증명 법 n의 원시근은 $(\mathbf{Z}/n\mathbf{Z})^*$의 생성원이다. 가정에 의하여 $(\mathbf{Z}/n\mathbf{Z})^*$은 위수가 $\varphi(n)$인 순환군이고, 원시근이 곧 순환군의 생성원이다. 따라서 원시근의 개수는 순환군의 생성원의 개수와 같다. 그런데 위수가 r인 순환군 $\langle g\rangle$의 원소 g^k의 위수가 r일 필요충분조건은 $\gcd(k,r)=1$이므로 그런 $1\le k\le r$의 개수는 $\varphi(r)$이다. 따라서, $(\mathbf{Z}/n\mathbf{Z})^*$은 위수가 $\varphi(n)$이므로, $\varphi(\varphi(n))$ 개의 원시근이 존재한다. □

예 2.5.13. 법 17은 $\varphi(\varphi(17))=\varphi(16)=2^4-2^3=8$개의 원시근을 갖는다. 모두 찾아보면, $3,5,6,7,10,11,12,14$이다. 법 9는 $\varphi(\varphi(9))=\varphi(6)=2$개의 원시근을 갖는데, 2와 5가 법 9의 원시근이다. 앞에서 확인하였듯이 8의 원시근은 없다.

2.5.3 Artin의 가설

가설 2.5.14 (Emil Artin)**.** *정수 $a\in\mathbf{Z}$는 -1도 아니고 제곱수도 아니라고 하자. 그러면 a가 원시근이 되는 소수 p는 무한히 많다.*

Artin의 가설을 만족하는 한 개의 a도 아직까지는 알려져 있지 않다. 임의의 정수 a에 대하여 Pieter[37]는 a의 위수가 $p-1$의 가장 큰 소수 약수에 의해 나누어지는 소수 p는 무한히 많이 존재한다는 것은 증명하였다. Hooley[22]는 소위 일반적인 리만 가설이라고 불리워지는 가설이 Artin의 추론 2.5.14을 유도한다는 것을 보였다.

참조 2.5.15. Artin의 가설을 좀 더 정확히 설명하면 a가 법 p의 원시근이 되는 x 이하의 소수 p들의 개수를 $N(x,a)$라고 하면, a에만 의존하는 양의 상수 $C(a)$가 존재하여 $N(x,a)$는 $C(a)\pi(x)$에 가까워진다는 것이다.

2.5.4 원시근의 계산

정리 2.5.8는 원시근을 계산하는 효과적인 알고리듬을 제시하지는 않는다. 법 p의 원시근을 실제 계산하기 위해서는, p가 정해지면 $a=2$가 원시근이 되는지 거듭제곱을 계산해본다. 실패하면 $a=3$으로 놓고 같은 계산을 반복하는데, 위수가 $p-1$인 a를 찾을 때 까지 이런 계산을 반복한다. $(\mathbf{Z}/p\mathbf{Z})^*$의 원소의 위수의 계산에는 $p-1$의 인수분해가 필요한데, 일반적으로 빠른 인수분해 방법은 알려져 있지않다. 따라서 큰 소수 p의 원시근을 찾는 일은 어려운 문제처럼 보인다.

알고리듬 2.5.16 (원시근)**.** 소수 p를 주면 이 알고리듬은 가장 작은 p의 원시근을 찾아준다.

1. [$p=2$?] 만약 $p=2$이면 1를 출력하고 끝난다. 그렇지 않으면 $a=2$로 놓는다.
2. [소수 약수들 찾기] $p-1$의 소수 약수 $p_1,\dots,p_r$을 계산한다.

3. [생성원?] 만약 모든 p_i에 대하여, $a^{(p-1)/p_i} \not\equiv 1 \pmod{p}$이면, a는 $(\mathbf{Z}/p\mathbf{Z})^*$를 생성한다. 따라서 a를 출력하고 끝난다.
4. [다음 수 확인] $a = a + 1$로 놓고 Step 3으로 간다.

증명 $a \in (\mathbf{Z}/p\mathbf{Z})^*$라고 하자. 군 $(\mathbf{Z}/p\mathbf{Z})^*$의 위수가 $p-1$이므로 a의 위수 d는 $p-1$의 약수이다. $d = (p-1)/n$ (n은 $p-1$의 약수)라 쓰자. 만약 a가 $(\mathbf{Z}/p\mathbf{Z})^*$의 생성원이 아니라면, $1 < n \mid (p-1)$이므로, $p_i \mid n$인 $p-1$의 소수 약수 p_i가 존재한다. 그러면

$$a^{(p-1)/p_i} = (a^{(p-1)/n})^{n/p_i} \equiv 1 \pmod{p}$$

가 성립한다. 역으로, a가 생성원이라면, $p-1$의 모든 소수 약수 p_i에 대해 $a^{(p-1)/p_i} \not\equiv 1 \pmod{p}$이다. 따라서 이 알고리듬은 Step 3에서 끝나기 위한 필요충분조건은 a가 원시근이다. 정리 2.5.8에 의하여, 적어도 한 개의 원시근은 존재하므로 이 알고리듬은 유한 단계에서 끝난다. □

2.6 Exercises

2.1 Prove that for any positive integer n, the set $(\mathbf{Z}/n\mathbf{Z})^*$ under multiplication modulo n is a group.

2.2 Compute the following gcd's using Algorithm 1.1.13:

$$\gcd(15, 35) \quad \gcd(247, 299) \quad \gcd(51, 897) \quad \gcd(136, 304)$$

2.3 Use Algorithm 2.3.7 to find x, $y \in \mathbf{Z}$ such that $2261x + 1275y = 17$.

2.4 Prove that if a and b are integers and p is a prime, then $(a+b)^p \equiv a^p + b^p \pmod{p}$. You may assume that the binomial coefficient

$$\frac{p!}{r!(p-r)!}$$

is an integer.

2.5 (a) Prove that if x, y is a solution to $ax + by = d$, with $d = \gcd(a, b)$, then for all $c \in \mathbf{Z}$,

$$x' = x + c \cdot \frac{b}{d}, \qquad y' = y - c \cdot \frac{a}{d} \tag{2.6.1}$$

is also a solution to $ax + by = d$.

(b) Find two distinct solutions to $2261x + 1275y = 17$.

(c) Prove that all solutions are of the form (2.6.1) for some c.

2.6 Let $f(x) = x^2 + ax + b \in \mathbf{Z}[x]$ be a quadratic polynomial with integer coefficients, for example, $f(x) = x^2 + x + 6$. Formulate a conjecture about when the set

$$\{f(n) : n \in \mathbf{Z} \text{ and } f(n) \text{ is prime}\}$$

is infinite. Give numerical evidence that supports your conjecture.

2.7 Find four complete sets of residues modulo 7, where the ith set satisfies the ith condition: (1) nonnegative, (2) odd, (3) even, (4) prime.

2.8 Find rules in the spirit of Proposition 2.1.9 for divisibility of an integer by 5, 9, and 11, and prove each of these rules using arithmetic modulo a suitable n.

2.9 (*) *(The following problem is from the 1998 Putnam Competition.)* Define a sequence of decimal integers a_n as follows: $a_1 = 0$, $a_2 = 1$, and a_{n+2} is obtained by writing the digits of a_{n+1} immediately followed by those of a_n. For example, $a_3 = 10$, $a_4 = 101$, and $a_5 = 10110$. Determine the n such that a_n is a multiple of 11, as follows:

(a) Find the smallest integer $n > 1$ such that a_n is divisible by 11.

(b) Prove that a_n is divisible by 11 if and only if $n \equiv 1 \pmod 6$.

2.10 Find an integer x such that $37x \equiv 1 \pmod{101}$.

2.11 What is the order of 2 modulo 17?

2.12 Let p be a prime. Prove that $\mathbf{Z}/p\mathbf{Z}$ is a field.

2.13 Find an $x \in \mathbf{Z}$ such that $x \equiv -4 \pmod{17}$ and $x \equiv 3 \pmod{23}$.

2.14 Prove that if $n > 4$ is composite then

$$(n-1)! \equiv 0 \pmod n.$$

2.15 For what values of n is $\varphi(n)$ odd?

2.16 (a) Prove that φ is multiplicative as follows. Suppose m, n are positive integers and $\gcd(m, n) = 1$. Show that the natural map $\psi : \mathbf{Z}/mn\mathbf{Z} \to \mathbf{Z}/m\mathbf{Z} \times \mathbf{Z}/n\mathbf{Z}$ is an injective homomorphism of rings, hence bijective by counting, then look at unit groups.

(b) Prove conversely that if $\gcd(m, n) > 1$, then the natural map $\psi : \mathbf{Z}/mn\mathbf{Z} \to \mathbf{Z}/m\mathbf{Z} \times \mathbf{Z}/n\mathbf{Z}$ is not an isomorphism.

2.17 Seven competitive math students try to share a huge hoard of stolen math books equally between themselves. Unfortunately, six books are left over, and in the fight over them, one math student is expelled. The remaining six math students, still unable to share the math books equally since two are left over, again fight, and another is expelled. When the remaining five share the books, one book is left over, and it is only after yet another math student is expelled that an equal sharing is possible. What is the minimum number of books that allows this to happen?

2.18 Show that if p is a positive integer such that both p and $p^2 + 2$ are prime, then $p = 3$.

2.19 Let $\varphi : \mathbf{N} \to \mathbf{N}$ be the Euler φ function.

(a) Find all natural numbers n such that $\varphi(n) = 1$.

(b) Do there exist natural numbers m and n such that $\varphi(mn) \neq \varphi(m) \cdot \varphi(n)$?

2.20 Find a formula for $\varphi(n)$ directly in terms of the prime factorization of n.

2.21 (a) Prove that if $\varphi : G \to H$ is a group homomorphism, then $\ker(\varphi)$ is a subgroup of G.

(b) Prove that $\ker(\varphi)$ is **normal**, i.e., if $a \in G$ and $b \in \ker(\varphi)$, then $a^{-1}ba \in \ker(\varphi)$.

2.22 Is the set $\mathbf{Z}/5\mathbf{Z} = \{0, 1, 2, 3, 4\}$ with binary operation multiplication modulo 5 a group?

2.23 Find all *four* solutions to the equation

$$x^2 - 1 \equiv 0 \pmod{35}.$$

2.24 Prove that for any positive integer n the fraction $(12n+1)/(30n+2)$ is in reduced form.

2.25 Suppose a and b are positive integers.

(a) Prove that $\gcd(2^a - 1,\ 2^b - 1) = 2^{\gcd(a,b)} - 1$.

(b) Does it matter if 2 is replaced by an arbitrary prime p?

(c) What if 2 is replaced by an arbitrary positive integer n?

2.26 For every positive integer b, show that there exists a positive integer n such that the polynomial $x^2 - 1 \in (\mathbf{Z}/n\mathbf{Z})[x]$ has at least b roots.

2.27 (a) Prove that there is no primitive root modulo 2^n for any $n \geq 3$.

(b) (*) Prove that $(\mathbf{Z}/2^n\mathbf{Z})^*$ is generated by -1 and 5.

2.28 Let p be an odd prime.

(a) (*) Prove that there is a primitive root modulo p^2. (Hint: Use that if a, b have orders n, m, with $\gcd(n, m) = 1$, then ab has order nm.)

(b) Prove that for any n, there is a primitive root modulo p^n.

(c) Explicitly find a primitive root modulo 125.

2.29 (*) In terms of the prime factorization of n, characterize the integers n such that there is a primitive root modulo n.

2.30 Compute the last two digits of 3^{45}.

2.31 Find the integer a such that $0 \leq a < 113$ and

$$102^{70} + 1 \equiv a^{37} \pmod{113}.$$

2.32 Find the proportion of primes $p < 1000$ such that 2 is a primitive root modulo p.

2.33 Find a prime p such that the smallest primitive root modulo p is 37.

3
공개키 암호 (Public-key Cryptography)

1970년 대에 정수론을 바탕으로 개발 된 기술 덕분에 제 3자가 중간에 내용을 모두 가로채 읽는다 하더라도 비밀 통신이 가능한 시대가 처음으로 도래하였다. 그리고 이 아이디어는 오늘날에도 여전히 유효하다. 사실 우리가 온라인 쇼핑을 할 때마다 법 n 정수들의 환에서 작동하는 이 시스템을 이용한다. 이 단원은 여러 가지 그런 시스템에 대한 이야기이다.

3.1 불놀이

나는 최근에 Section One이라고 불리는 음울한 반 테러 비밀 기관의 첩보원이 될 수 밖에 없었던 Nikita라는 이름을 가진 여성에 관한 La Femme Nikita라는 텔레비젼 쇼를 시청하였다. 이 쇼에서 Nikita는 동료 첩보원 Michael에 대하여 강한 감정을 가지고 있고, 폭발 전문가인 Section One의 Walter를 가장 신뢰한다. 가끔은 Section One 안에 있는 그녀의 상관과 동료들이 Nikita의 최악의 적이기도 하다. 다음은 3부의 에피소드 개요이다.

불장난

중무장한 테러 단체의 베이스 캠프로부터 폭발 칩을 제거하기 위한 특명을 수행 중에 Nikita는 인질이 된다. 적어도 그렇게 보인다. 사실 Michael와 Nikita는 비밀리에 만나기 위하여 함께 시나리오를 만들었다. 이 계략은 작동하지만 Section One의 전문 해커인 Birkoff가 Michael과 Nikita가 Walter의 도움으로 주고 받은 암호 통신문을 우연히 발견하고 Madeline에게 말할 수 밖에 없는 상황이 된다. Section One과 Madeline은 Michael와 Nikita가 배신 할

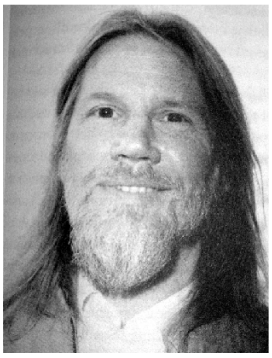
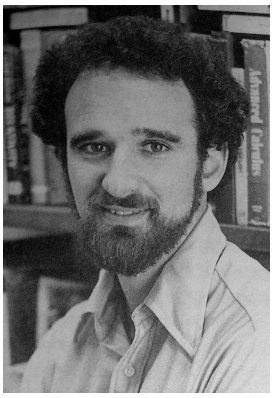



FIGURE 3.1. Diffie and Hellman (photos from [**?**])

> 수도 있다고 의심하면서 이들의 비밀 만남을 추적하기 위하여 ...
> 필요하다면 그 들을 죽일 수도 있는... 두 번 째 팀을 만든다.

Walter는 어떤 종류의 암호를 그들이 사용하도록 도왔을까? 내가 자유롭게 상상한 후 떠 오른 것은 Nikita가 베이스 캠프에서 인질로 잡혀을 때 납치범의 전화기를 얻어 그녀에게 어떤 일이 일어났는지를 동료들이 알아낼 수 있도록 전화기를 이용했을 것 같다는 것이다. 모두가 열심히 그녀의 전화를 듣고 있다.

참조 3.1.1. 이 책에서는 정수를 무작위로 선택하는 (난수 생성) 방법을 (여러분들이) 이용할 수 있다고 가정한다. 실제 이 방법은 존재하고 흥미롭지만 이 책에서는 다루지 않는다. [28, Ch. 3]를 참고한다.

Nikita는 엿듣는 사람이 있을 때에도 비밀키를 정할 수 있다는 "Diffie-Hellman key exchange"라는 공개키 암호에 관한 Walter와의 대화를 기억해 낸다. 게다가 Walter는 비록 Diffie-Hellman이 최초의 공개키 암호지만 오늘날도 여전히 사용되고 있다고 언급하였다 (예를 들어, OpenSSH 프로토콜 버전 2에서, `http://www.openssh.com/` 참조).

니키타는 그녀의 휴대용 컴퓨터와 휴대 전화를 꺼내서 Michael을 호출하고, *오류가 있는* 다음 과정을 수행한다. (독자들은 읽으면서 무엇이 틀렸는지 찾아보기 바란다.)

1. 두 사람은 함께 큰 소수 p와 $1 < g < p$를 만족하는 정수 g를 선택한다.
2. Nikita는 *비밀리에* 정수 n을 선택한다.
3. Michael도 *비밀리에* 정수 m을 선택한다.
4. Nikita는 Michael에게 $ng \pmod{p}$를 말한다.
5. Michael은 Nikita에게 $mg \pmod{p}$를 말한다.

6. 비밀키는 $s = nmg \pmod{p}$인데 이 수는 Nikita와 Michael 모두 쉽게 계산할 수 있다.

다음 예는 Nikita와 Michael이 한 일들을 작은 수를 가지고 예시한 것이다. (실제 그들은 훨씬 더 큰 수를 사용하였다.)

1. $p = 97$, $g = 5$
2. $n = 31$
3. $m = 95$
4. $ng \equiv 58 \pmod{97}$
5. $mg \equiv 87 \pmod{97}$
6. $s = nmg \equiv 78 \pmod{97}$

그런데 모두가 쉽게 s를 알아낼 수 있어 Nikita와 Michael는 들키고 만다.

1. 모든 사람들은 p, g, $ng \pmod{p}$, 그리고 $mg \pmod{p}$를 알고 있다.
2. $\gcd(g, p) = 1$이기 때문에, 알고리듬 2.3.7이용하면, 누구든지 $ag + bp = 1$인 두 정수 $a, b \in \mathbf{Z}$를 찾을 수 있다.
3. 그러면 $ang \equiv n \pmod{p}$이므로, a와 ng를 알고 있으면 Nikita의 비밀키인 n을 계산할 수 있고, 따라서 두 사람의 공동키인 s를 구할 수 있다.

Nikita를 조롱하려고 Nikita의 납치범은 Diffie와 Hellman의 1976 논문인 "New Directions in Cryptography" [16]의 평론의 일부를 보여준다:

> "저자들은 통신이론의 최신 결과를 논한다. 첫번째 방법의 특징은 불법으로 정보를 도청하더라도 계산적으로 해독이 거의 불가능하며 이 시스템을 실행하기 위한 두 세 가지 기술을 제안하고 있다. 그러나 평론가들은 확신을 하지 못한다.

3.2 Diffie-Hellman의 열쇠 교환

니키타의 방에도 어둠이 몰려올 때, 니키타는 어떤 일들이 일어났는지를 다시 생각한다. 그리고 잘못 기억한 것을 떠올리고 미카엘에게 전화를 걸어 두 사람은 다음과 같은 작업을 한다.

1. Michael과 Nikita는 함께 소수일 것 같은 200-자릿수의 정수 p를 선택하고, 또 1과 p사이의 정수 g를 선택한다.
2. Nikita는 정수 n을 선택하고 *혼자만 기억한다.*

3. Michael는 정수 m을 선택하고 *혼자만 기억한다*.

4. Nikita는 그녀의 컴퓨터에서 $g^n \pmod p$을 계산하여 그 값을 Michael에게 전화로 알려준다.

5. Michael은 Nikita에게 $g^m \pmod p$을 알려준다.

6. 이 두사람의 암호키는

$$s \equiv (g^n)^m \equiv (g^m)^n \equiv g^{nm} \pmod p,$$

이며, 이 값은 두 사람만이 계산할 수 있다.

다음은 두 사람이 한 작업을 상대적으로 간단한 계산으로 보여주는 예이다.

1. $p = 97$, $g = 5$
2. $n = 31$
3. $m = 95$
4. $g^n \equiv 7 \pmod p$
5. $g^m \equiv 39 \pmod p$
6. $s \equiv (g^n)^m \equiv 14 \pmod p$

3.2.1 이산로그문제

이제 Nikita는 Michael과 모든 대화를 (예를 들어 AES, Arcfour, Cast128, 3DES, 혹은 Blowfish와 같은 표준 대칭 암호를 사용하여) 두 사람의 비밀키를 이용하여 암호화하여 통신한다. 그들의 대화를 해독하려면 도청자는 s를 알아야 한다. 그러나 알려진 정보 p, g, g^n과 g^m로부터 s를 계산하는 것는 아주 시간이 많이 걸리는 작업이다. 한 방법은 g와 g^n으로부터 n을 구하는 것인데 현재로는 "계산적으로 실행불가능"한 것으로 알려져있다. "계산적으로 실행불가능"하다는 것은 계산하는데 시간이 너무 오래 걸려 실용적이지 못하다는 의미이다.

a, b는 양의 실수이고 n은 실수일 때 밑수가 b인 log 함수는

$$\log_b(a) = n \quad \Longleftrightarrow \quad a = b^n.$$

로 정의되었음을 기억하자. 대수학에서 $\log_b$ 함수는

$$a = b^n$$

을 만족하는 지수 n을 구하기 위한 함수이다. 즉, $a = b^n$과 b가 주어졌을 때 $n = \log_b(a)$이다.

SAGE 예 3.2.1. $a = 19683$는 $b = 3$의 n번 거듭제곱이다. 즉, $3^n = 19683$. 그러면

$$n = \log_3(19683) = \log(19683)/\log(3) = 9$$

이다. 이 값은 Sage에서 쉽게 구할 수 있다.

```
sage: log(19683.0)
9.88751059801299
sage: log(3.0)
1.09861228866811
sage: log(19683.0) / log(3.0)
9.00000000000000
```

Sage는 모든 x에 대하여 $\log(x)$의 근사값을 (적어도 어떤 x 범위에서는) 빨리 수렴하는 적절한 무한 급수를 이용하여 빨리 구할 수 있다.

이산로그문제(discrete log problem)는 $\log_b(a)$를 계산하는 것과 유사하지만 여기서 b와 a가 유한군의 원소이다.

문제 3.2.2 (이산로그문제)**.** 집합 G는 예를 들어 $(\mathbf{Z}/p\mathbf{Z})^*$와 같은 유한군이다. $b \in G$이고 a가 b의 거듭제곱일 때, $b^n = a$을 만족하는 양의 정수 n을 구하여라.

우리가 아는 한 p가 상당히 큰 소수일 때 $(\mathbf{Z}/p\mathbf{Z})^*$에서의 이산로그문제는 실제로 아주 어려운 문제이다. 오랫동안 많은 사람들이 큰 동기를 부여 받아 해결하려고 노력하였다. 예를 들어 만약 니키타의 납치범이 3.2.2 문제를 효율적으로 풀었더라면 그들은 니키타와 미카엘의 통신 내용을 읽을 수 있었을 것이다. 불행히도 현재 우리가 사용하는 컴퓨터에서 이산로그문제가 아주 어렵다는 것은 증명되지는 않았다. 또 Peter Shor는 [46]에서 우리가 충분히 복잡한 양자 컴퓨터를 만들수만 있다면 이산로그문제가 $\#G$의 자릿수의 다항식 함수로 주어지는 시간 안에 풀릴 수 있음을 보였다.

이산로그문제의 비효율적인 알고리듬을 주는 것은 쉽다. 그냥 $b^n = a$가 나올 때까지 b^1, b^2, b^3 등을 계속 계산하는 것이다. 예를 들어 $a = 18$, $b = 5$, $p = 23$이라고 가정하고

$$b^1 = 5,\ b^2 = 2,\ b^3 = 10,\ \ldots,\ b^{12} = 18,$$

을 계산하여 $n = 12$를 찾는 것이다. 그러나 p가 아주 큰 경우 이런 방법으로 이산로그문제를 푸는 것은 실용적이지 않다. 왜냐하면 숫자가 커지면 계산하는데 어마아마하게 긴 시간이 필요하기 때문이다.

SAGE 예 3.2.3. 아마도 이산로그 계산이 어려운 이유 중의 일부는 실수에서의 로그함수는 연속인데 법 n에 관한 로그는 임의로 값이 큰 폭으로 달라진다는 점일 수 있다. 이 색다른 현상을 그림 3.2에서 관찰하자.

다음 코드는 그림을 연속적으로 그린다.

```
sage: plot(log, 0.1,10, rgbcolor=(0,0,1))
```

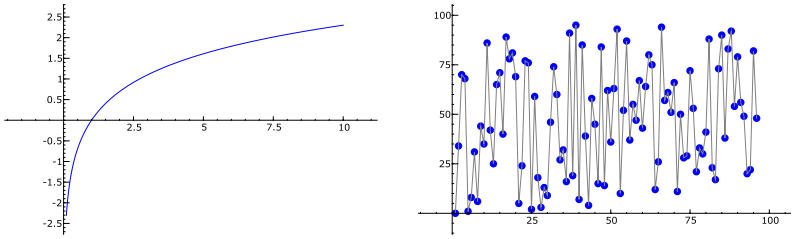


FIGURE 3.2. Graphs of the continuous log and of the discrete log modulo 53. Which picture looks easier to predict?

이번은 비연속적인 그림을 그린다.

```
sage: p = 53
sage: R = Integers(p)
sage: a = R.multiplicative_generator()
sage: v = sorted([(a^n, n) for n in range(p-1)])
sage: G = plot(point(v,pointsize=50,rgbcolor=(0,0,1)))
sage: H = plot(line(v,rgbcolor=(0.5,0.5,0.5)))
sage: G + H
```

3.2.2 현실적인 Diffie-Hellman 키 교환의 예

이 절에서는 더 큰 수를 사용하는 예를 제시한다. 먼저 우리는 $(\mathbf{Z}/p\mathbf{Z})^*$에서 위수가 $p-1$이 되는 원소 g를 찾기 쉬운 소수 p를 선택할 수 있다는 기초정리를 증명한다. 이미 2.5절에서 모든 소수 p는 법 p의 원시근이 존재한다는 것을 증명하였고 그런 원시근을 찾는 알고리듬도 공부하였다. 아래 기초정리 3.2.4의 중요성은 p가 큰 수일 때, $p-1$의 인수분해를 요구하지 않기 때문에, 실제 사용하기 더 적절한 원시근을 찾아주는 방법을 제시한다는 것이다. 물론 Diffie-Hellman을 위해서라면 g가 원시근일 필요는 없으므로 임의로 선택하여도 무방하다.

기초정리 3.2.4. *p는 소수이고 또 $(p-1)/2$도 소수라고 하자. 그러면 $(\mathbf{Z}/p\mathbf{Z})^*$의 각각의 원소는 위수가 1, 2, $(p-1)/2$, 혹은 $p-1$이다.*

증명 p는 소수이므로 $(\mathbf{Z}/p\mathbf{Z})^*$는 위수가 $p-1$인 군이다. 가정에 의하여 $p-1$은 $2\cdot((p-1)/2)$로 소인수분해가 된다. $a\in(\mathbf{Z}/p\mathbf{Z})^*$이면 정리 2.1.20에 의하여 $a^{p-1}=1$이 되고, 이로부터 a의 위수는 $p-1$의 약수이다. 2와 $(p-1)/2$이 $p-1$의 소수 약수이므로, a의 위수는 1, 2, $(p-1)/2$, 혹은 $p-1$이다. □

법이 소수 p일 때 $(p-1)/2$이 소수가 되면 위수가 $p-1$인 원소를 다음과 같이 찾을 수 있다. 먼저 2의 위수를 확인한다. 2의 위수가 $p-1$이면 2가 우리가 원하는 것이다. 2가 아니라면, 2의 위수는 1이나 2는 결코 아니므로 2의 위수는 $(p-1)/2$이고 따라서 -2의 위수가 $p-1$이 된다.

$p = 934509830948509384509834096l1$라 놓자. 그러면 p는 소수이지만 $(p-1)/2$은 소수가 아니다. 따라서 p에 2를 계속 더하면서 절 2.4에 있는 유사소수 판정을 계속하여 p 다음의 유사소수를 찾는다.

$$q = 93450983094850938450983409623.$$

는 유사소수이고 $(q-1)/2$도 유사소수이다. 이제, q가 진짜 소수라고 가정한다면, 2의 위수가 $(q-1)/2$임을 확인하여 $g = -2$가 법 q에 관하여 위수가 $q-1$임을 안다.

Nikita와 Michae가 생성한 비밀키는 각각

$$n = 183199223755318591716133791 81$$

이고

$$m = 8233583624386669568014144030 0$$

이다. 따라서 Nikita는

$$g^n = 45416776270485369791375944998 \in (\mathbf{Z}/q\mathbf{Z})^*$$

을 Michael에게 보내고, 그리고 Michael은

$$g^m = 150480741517708842718242253 93 \in (\mathbf{Z}/q\mathbf{Z})^*$$

을 Nikita에게 각각 보낸다. 그들이 함께 공유한 비밀키는

$$g^{nm} = 85771409470770521212346739540 \in (\mathbf{Z}/q\mathbf{Z})^*$$

이다.

SAGE 예 3.2.5. Sage를 이용하여 위에서 설명한 계산을 수행한 예시이다.

```
sage: q = 93450983094850938450983409623
sage: q.is_prime()
True
sage: is_prime((q-1)//2)
True
sage: g = Mod(-2, q)
sage: g.multiplicative_order()
93450983094850938450983409622
sage: n = 183199223755318591716133791 81
sage: m = 8233583624386669568014144030 0
sage: g^n
45416776270485369791375944998
sage: g^m
150480741517708842718242253 93
sage: (g^n)^m
85771409470770521212346739540
sage: (g^m)^n
85771409470770521212346739540
```

3.2.3 중간 공격수

두 사람의 첫 번째 시스템이 실패하였으므로 전화기로 통화하는 대신 Michael과 Nikita는 문자로만 통신할 수 있다. 납치자 중 한 사람인 The Man이 이들의 문자 통신을 모두 지켜본다. 그뿐만 아니라 문자를 중간에 가로채어 다른 문자를 보낸다. Nikita가 Michael에게 $g^n \pmod p$을 문자할 때 The Man은 이 문자를 가로챈 후 자신의 수인 $g^t \pmod p$를 Michael에게 보낸다. 결론적으로 Michael과 The Man은 $g^{tm} \pmod p$이라는 비밀키를 공유하게 되고 Nikita와 The Man은 $g^{tn} \pmod p$이라는 비밀키를 공유하게 된다. Nikita가 Michael에게 문자를 보낼 때 비밀키 $g^{tn} \pmod p$을 이용하게 되고, The Man은 이를 가로채 복호화하고 내용을 바꾼 후 $g^{tm} \pmod p$을 사용하여 바꾼 내용을 다시 암호화하여 Michael에게 보낸다. 이제 The Man은 Michael과 Nikita 사이의 모든 문자를 읽을 수 있고 심지어 내용까지 바꿀 수 있는 나쁜 경우가 된다.

이 공격을 피할 수 있는 한 가지 방법은 RSA 암호를 바탕으로 한 전자 서명 scheme을 사용하는 것이다. 전자서명에 대해서는 이 책에서 더 이상 논하지는 않을 것이지만 RSA는 다음 절에서 논하려고 한다.

3.3 RSA 암호

Diffie-Hellman의 키교환에는 단점이 있다. 절 3.2.3에서 논하였듯이, 중간에 있는 도청자에 의해 영향을 받을 수 있다. 이 절에서는 Diffie-Hellman의 대안으로 Rivest, Shamir, 그리고 Adleman의 RSA 공개키 암호를 소개한다[43]. RSA 공개키 암호는 사용하기에 좀 더 적절하다고 알려져 있다.

먼저 RSA 암호를 설명하고, 이 암호를 공격할 여러 방법에 대해 논한다. RSA암호를 사용하면서 어리석은 실수를 하지 않기 위해서는 이 암호의 여러 약점들을 알고 있어야 한다. 그러나 이 책에서는 RSA나 다른 암호들의 특정한 실행에 대한 가능한 공격들에 대해서는 거의 다루지 못하였다.

3.3.1 RSA 작동 원리

RSA의 기본 아이디어는 어떤 집합 X에 trap-door 혹은 일방향 함수(one-way function)을 만드는 것이다. 이 함수

$$E : X \to X$$

는 가역함수인데 역함수 E^{-1}를 Nikita는 쉽게 계산할 수 있으나 나머지 사람들은 계산하기가 아주 어려운 함수이다.

Nikita가 법 n에 관한 정수들의 집합에 일방향 함수 E를 어떻게 만드는지를 여기에 설명한다.

1. 2.4절에 암시한 방법을 사용하여, Nikita는 두 개의 큰 소수 p와 q를 선택하고, $n = pq$로 놓는다.

2. 그러면 Nikita는 쉽게 $\varphi(n)$을 계산한다:

$$\varphi(n) = \varphi(p) \cdot \varphi(q) = (p-1) \cdot (q-1).$$

3. 다음 Nikita는

$$1 < e < \varphi(n) \text{와 } \gcd(e, \varphi(n)) = 1$$

를 만족하는 정수 e를 무작위로 선택한다.

4. Nikita는 절 2.3.2의 알고리듬을 사용하여

$$ex \equiv 1 \pmod{\varphi(n)}.$$

의 해 $x = d$를 찾는다.

5. 마지막으로 Nikita는 함수 $E : \mathbf{Z}/n\mathbf{Z} \to \mathbf{Z}/n\mathbf{Z}$를

$$E(x) = x^e \in \mathbf{Z}/n\mathbf{Z}.$$

로 정의한다.

누구든 절 2.3.2의 거듭제곱알고리듬을 이용하면 E를 상당히 빨리 계산할 수 있다. Nikita의 **공개키(public key)**는 두 정수쌍 (n, e)이며, 이는 사람들이 E를 쉽게 계산하기 위하여 꼭 필요한 정보이다. Nikita는 $ed \equiv 1 \pmod{\varphi(n)}$을 만족하는 d를 알고 있으므로, 곧 우리도 확인하겠지만, E^{-1}을 쉽게 계산할 수 있다.

Nikita에게 평문을 보내기 위해서는 다음과 같이 진행한다. 평문을 법 n에 관한 정수들의 수열 (절 3.3.2 참조)

$$m_1, \ldots, m_r \in \mathbf{Z}/n\mathbf{Z},$$

으로 코딩한 후

$$E(m_1), \ldots, E(m_r)$$

을 Nikita에게 보낸다. ($E(m) = m^e$ ($m \in \mathbf{Z}/n\mathbf{Z}$)임을 상기하도록.)

Nikita는 $E(m_i)$를 받아 $E^{-1}(m) = m^d$ 와 기초정리 3.3.1로 부터 유도된 사실을 이용하여 m_i를 구한다.

기초정리 3.3.1 (복호키(Decryption Key)). *n은 서로 다른 소수들의 곱이고, 정수 $d, e \in \mathbf{N}$는 $p \mid n$인 소수 p에 대하여 $p-1 \mid de-1$ 을 만족하는 정수이다. 그러면 모든 $a \in \mathbf{Z}$에 대하여 $a^{de} \equiv a \pmod{n}$이 성립한다.*

증명 $n \mid a^{de} - a$는 $p \mid n$인 모든 소수 p가 $p \mid a^{de} - a$인 것과 동치이므로 n의 모든 소수 약수 p에 대하여 $a^{de} \equiv a \pmod{p}$임을 보이면 충분하다. $\gcd(a, p) \neq 1$이면 $a \equiv 0 \pmod{p}$이므로 $a^{de} \equiv a = 0 \pmod{p}$가 성립한다. $\gcd(a, p) = 1$이면 정리 2.1.20에 의해 $a^{p-1} \equiv 1 \pmod{p}$가 성립한다. $p - 1 \mid de - 1$이므로 $a^{de-1} \equiv 1 \pmod{p}$도 성립한다. 양변을 a로 곱하면 $a^{de} \equiv a \pmod{p}$이다. □

따라서 $E(m_i)$를 복호화 하기 위하여 Nikita는 다음을 계산한다.

$$E(m_i)^d = (m_i^e)^d = m_i.$$

SAGE 예 3.3.2. Sage를 이용하여 RSA암호를 실행한다. `rsa` 함수는 bit를 정해주면 (최대) 그 bit가 되는 키를 만든다. 즉, `bits`가 20이면 $n = pq$인 키를 생성하는데 이 수의 크기는 약 2^{20}정도이다. 실제 사용되는 RSA 키의 크기는 512, 1024, 혹은 2048 bit이다. Sage를 이용하여 큰 키들을 생성해보고 시간이 얼마나 걸리는지 확인해보자.

(다음은 Sage에 RSA 키와 암호화, 복호화 함수를 정의하는 방법을 알려주고 있다. 예를 들어 rsa(20)을 입력 후 계산하면 (7177, 13753, 33169) 를 출력한다. 암호화 연습도 해보길 바란다.)

```
sage: def rsa(bits):
...    # only prove correctness up to 1024 bits
...    proof = (bits <= 1024)
...    p = next_prime(ZZ.random_element(2**(bits//2 +1)),
...                proof=proof)
...    q = next_prime(ZZ.random_element(2**(bits//2 +1)),
...                proof=proof)
...    n = p * q
...    phi_n = (p-1) * (q-1)
...    while True:
...        e = ZZ.random_element(1,phi_n)
...        if gcd(e,phi_n) == 1: break
...    d = lift(Mod(e,phi_n)^(-1))
...    return e, d, n
...
sage: def encrypt(m,e,n):
...    return lift(Mod(m,n)^e)
...
sage: def decrypt(c,d,n):
...    return lift(Mod(c,n)^d)
...
sage: e,d,n = rsa(20)
sage: c = encrypt(123, e, n)
sage: decrypt(c, d, n)
123
```

3.3.2 문장을 숫자로 코딩하기

RSA암호를 사용하여 평문을 암호문으로 만들기 위하여 먼저 평문을 $n = pq$보다 작은 수들의 열로 **코딩**(부호화 하는 것이)하는 것이 필요하다. 지금 이 방법을 간단히 설명하려 한다. 모든 실제 구현에서는 여분의 무작위로 선택한 기호(마치 음식의 소금같은 역할을 해서 "salt"라고 칭함)를 모든 평문의 각

단위(block)의 제일 앞에 더하여 똑 같은 평문이라도 암호화할 때마다 새로운 암호문이 나오도록 한다. 이 방법은 선택한 평문의 공격을 막는 데 도움이 된다.

이제 s가 알파벳 대문자와 여백(space)의 수열이고 여백으로 시작하지 않는다고 가정하자. 그러면 여백은 0, A는 1, B는 2, 그리고 Z는 26으로 바꾸어 s를 27진법의 수로 코딩한다. 그러면 "RUN NIKITA"는 27진법의 다음 수로 표현된다.

$$\begin{aligned}\text{RUN NIKITA} \quad \leftrightarrow \quad & 27^9 \cdot 18 + 27^8 \cdot 21 + 27^7 \cdot 14 + 27^6 \cdot 0 + 27^5 \cdot 14 \\ & + 27^4 \cdot 9 + 27^3 \cdot 11 + 27^2 \cdot 9 + 27 \cdot 20 + 1 \\ & = 143338425831991 \text{ (in decimal).}\end{aligned}$$

이 숫자로부터 원래의 문장을 찾으려면 27로 계속 나누어 나머지에 해당하는 글자들로 바꾸어 읽으면 된다.

$$\begin{array}{rcrcrl}
143338425831991 & = & 5308830586370 \cdot 27 & + & 1 & \text{“A”} \\
5308830586370 & = & 196623355050 \cdot 27 & + & 20 & \text{“T”} \\
196623355050 & = & 7282346483 \cdot 27 & + & 9 & \text{“I”} \\
7282346483 & = & 269716536 \cdot 27 & + & 11 & \text{“K”} \\
269716536 & = & 9989501 \cdot 27 & + & 9 & \text{“I”} \\
9989501 & = & 369981 \cdot 27 & + & 14 & \text{“N”} \\
369981 & = & 13703 \cdot 27 & + & 0 & \text{“ ”} \\
13703 & = & 507 \cdot 27 & + & 14 & \text{“N”} \\
507 & = & 18 \cdot 27 & + & 21 & \text{“U”} \\
18 & = & 0 \cdot 27 & + & 18 & \text{“R”}
\end{array}$$

만약 $27^k \le n$이면 k개의 글자들은 위의 예시한 것처럼 한 번에 코딩한다. 따라서 만약 n이하인 정수들을 암호화 할 수 있다면 우리의 평문을 크기가 최대 $\log_{27}(n)$가 되도록 **블록(block)**들로 나누어야만 한다.

SAGE 예 3.3.3. 앞서 Sage를 이용하여 평문과 숫자 사이의 전환을 구현하기 위하여 밑수를 27로 잡았다. 이는 실제 구현할 때와 비교하면 마치 장난감 놀이 같은 예이다. 컴퓨터에서 입력 평문 s는 ASCII라 불리는 형식으로 저장되고 각 글짜(letter)는 0과 255 사이의 정수와 대응된다. 이 정수는 명령어 `ord`를 이용하여 얻어진다.

```
sage: def encode(s):
...     s = str(s)      # make input a string
...     return sum(ord(s[i])*256^i for i in range(len(s)))
sage: def decode(n):
...     n = Integer(n)  # make input an integer
...     v = []
...     while n != 0:
...         v.append(chr(n % 256))
...         n //= 256    # this replaces n by floor(n/256).
...     return ''.join(v)
```

```
sage: m = encode('Run Nikita!'); m
403547690147146494219687 22
sage: decode(m)
'Run Nikita!'
```

3.3.3 RSA암호의 완전한 예시

계산을 간단하게 하기 위하여 작은 소수 p와 q를 사용하여 RSA암호에서 한 글자 X를 암호화한다. 먼저 RSA암호의 매개변수들을 계산한다.

1. p와 q를 선택: $p = 17$, $q = 19$로 선택하면 $n = pq = 323$.
2. $\varphi(n)$을 계산:
$$\begin{aligned}\varphi(n) &= \varphi(p \cdot q) = \varphi(p) \cdot \varphi(q) = (p-1)(q-1) \\ &= pq - p - q + 1 = 323 - 17 - 19 + 1 = 288.\end{aligned}$$
3. 무작위로 $e < 288$이고 $\gcd(e, 288) = 1$인 수 선택 : $e = 95$를 선택하자.
4. GCD 알고리듬을 이용하여
$$95x \equiv 1 \pmod{288}$$
의 해 $d = 191$를 구한다.

위에서 RSA 공개키 암호의 매개변수들을 계산하였다. 공개키는 $(323, 95)$이며, 따라서 암호화 함수는
$$E(x) = x^{95}$$
로 정의되고, 복호화 함수는 $D(x) = x^{191}$이다.

다음 X를 암호화하는데 먼저 X를 숫자 24로 코딩하고
$$E(24) = 24^{95} = 294 \in \mathbf{Z}/323\mathbf{Z}$$
를 계산한다. 복호화 하기위해서는 E^{-1}를 계산:
$$E^{-1}(294) = 294^{191} = 24 \in \mathbf{Z}/323\mathbf{Z}.$$

다음 예는 더 큰 수들로 RSA 암호 과정을 예시한다.

$p = 738873402423833494183027176953$, $q = 37877768068656628823782 73$.

라 놓고자. 그러면

$n = p \cdot q = 279868753691091597012726360634791146094855419785 3542169$,

$$\begin{aligned}\varphi(n) &= (p-1)(q-1) \\ &= 279868753691091597012726286747072126030819435194 3986944.\end{aligned}$$

컴퓨터에 있는 난수발행기를 이용하여 저자는 정수 e를 선택한다.

$$e = 14839591948662041793485360102847166554421390249157206 99.$$

그러면

$$d = 21133679284963054695413483870886329734578023587816108 03.$$

$\log_{27}(n)$이 약 38.04이므로 38 글자를 한 묶음의 수로 코딩하고 암호화 할 수 있다. `RUN NIKITA`의 코드는 $m = 143338425831991$이므로 암호문은 다음과 같다.

$$\begin{aligned} E(m) &= m^e \\ &= 150455443299656813339308887860094810177372680087887399 0. \end{aligned}$$

참조 3.3.4. 실제로는 e는 작은 수로 선택하는데, e가 작다고 해도 RSA의 안전성이 약해지지는 않기 때문이다. 예를 들어 RSA의 실행에 대한 OpenSSL 문서에 의하면 (`http://www.openssl.org/` 참조) "The exponent is an odd number, typically 3, 17 or 65537." 으로 밝히고 있다.

3.4 RSA 공격하기

Nikita's의 공개키는 (n, e)이고 그녀의 복호화 키는 d라고 가정하면 $ed \equiv 1 \pmod{\varphi(n)}$을 만족한다. n을 pq로 소인수분해할 수 있으면 $\varphi(n) = (p-1)(q-1)$을 계산할 수 있고, 따라서 d도 구할 수 있다. 따라서 n을 인수분해할 수 있으면 n이 공개된 RSA암호를 깰 수 있다.

3.4.1 $\phi(n)$을 알 때 n을 인수분해하기

두 소수의 곱인 수 n의 $\varphi(n)$을 알면 $n = pq$를 만족하는 두 소수 p, q를 구하는 것은 쉽다.

$$\varphi(n) = (p-1)(q-1) = pq - (p+q) + 1$$

이고 $pq = n$이므로 $p + q = n + 1 - \varphi(n)$이 성립한다. 따라서 p와 q는 이차방정식

$$x^2 - (p+q)x + pq = (x-p)(x-q) = 0$$

의 해가 된다. 이 해들은 근의 공식으로도 찾을 수 있다.

예 3.4.1. $n = pq = 31615577110997599711$은 두 소수의 곱이고 $\varphi(n) = 31615577098574867424$이다. 따라서 p와 q는 다음 식으로부터 얻어진다.

$$\begin{aligned} f &= x^2 - (n + 1 - \varphi(n))x + n \\ &= x^2 - 12422732288x + 31615577110997599711 \\ &= (x - 3572144239)(x - 8850588049). \end{aligned}$$

마지막 단계의 인수분해는 다음 근의 공식으로 얻는다.

$$\begin{aligned}&\frac{-b+\sqrt{b^2-4ac}}{2a}\\ &\quad=\frac{12422732288+\sqrt{12422732288^2-4\cdot 31615577110997599711}}{2}\\ &\quad=8850588049.\end{aligned}$$

따라서 $n = 3572144239 \cdot 8850588049$임을 알 수 있다.

SAGE 예 3.4.2. 다음 Sage함수는 n과 $\varphi(n)$이 주어졌을 때 n을 $n = pq$로 인수분해한다.

```
sage: def crack_rsa(n, phi_n):
...     R.<x> = PolynomialRing(QQ)
...     f = x^2 - (n+1 -phi_n)*x + n
...     return [b for b, _ in f.roots()]
sage: crack_rsa(31615577110997599711, 31615577098574867424)
[8850588049, 3572144239]
```

3.4.2 p와 q가 가까울 때

두 소수 p와 q의 차가 작으면 **페르마의 인수분해법**(Fermat's factorization method)이라 불리는 페르마의 인수분해 방법을 쓰면 n을 쉽게 인수분해할 수 있다.

$n = pq$이고 $p > q$라고 가정하자. 그러면

$$n=\left(\frac{p+q}{2}\right)^2-\left(\frac{p-q}{2}\right)^2.$$

p와 q가 가까운 수이므로

$$s=\frac{p-q}{2}$$

는 작은 수이고

$$t=\frac{p+q}{2}$$

는 $\sqrt{n}$보다 약간 큰 수이면서 $t^2 - n = s^2$으로 제곱수이다. 따라서

$$t=\lceil\sqrt{n}\rceil,\quad t=\lceil\sqrt{n}\rceil+1,\quad t=\lceil\sqrt{n}\rceil+2,\ldots$$

등으로 놓고 $t^2 - n$이 완전 제곱수 s^2가 될 때 까지 계속 계산한다. (여기서 $\lceil x\rceil$ 은 x보다 같거나 큰 정수 중 가장 작은 정수이다. 그러면

$$p=t+s,\qquad q=t-s$$

가 된다.

예 3.4.3. $n = 23360947609$이다. 그러면

$$\sqrt{n} = 152842.88\ldots$$

이다.

$t = 152843$이면 $\sqrt{t^2 - n} = 187.18\ldots$.
$t = 152844$이면 $\sqrt{t^2 - n} = 583.71\ldots$.
$t = 152845$이면 $\sqrt{t^2 - n} = 804 \in \mathbf{Z}$.
따라서 $s = 804$이므로 $p = t + s = 153649$, $q = t - s = 152041$이다.

SAGE 예 3.4.4. $n = pq$이고 p와 q 중의 하나는 $\sqrt{n}$에 가까운 경우, 위의 인수분해 알고리듬을 실행하자.

```
sage: def crack_when_pq_close(n):
...    t = Integer(ceil(sqrt(n)))
...    while True:
...       k = t^2 - n
...       if k > 0:
...          s = Integer(int(round(sqrt(t^2 - n))))
...          if s^2 + n == t^2:
...             return t+s, t-s
...
...       t += 1
...
sage: crack_when_pq_close(23360947609)
(153649, 152041)
```

예를 들어 사람들은 무작위로 소수 하나를 찾은 후 그 다음 소수를 두 번 째 소수로 선택하면 좋을 꺼라고 생각할 수 있는데 그러면 금방 해독할 수 있는 암호가 된다.

SAGE 예 3.4.5.
```
sage: p = next_prime(2^128); p
340282366920938463463374607431768211507
sage: q = next_prime(p)
sage: crack_when_pq_close(p*q)
(340282366920938463463374607431768211537,
      340282366920938463463374607431768211507)
```

3.4.3 d를 알고 n을 인수분해하기

이 절에서는 RSA 암호에서 복호화 키 d가 주어졌을 때 n을 인수분해하는 확률적 알고리듬을 소개한다. 이 사실은 RSA 암호에서 복호화 키를 찾아내는 것은 적어도 실제 이용하는 입장에서는 계산적으로 n을 인수분해하는 것만큼 어려운 일이라는 것을 의미한다.

법 n과 암호화 키 e를 갖는 RSA 암호를 생각하자. 어떻게 하여 모든 a에 대해

$$a^{ed} \equiv a \pmod{n}$$

을 만족하는 d를 찾았다고 하자. 그러면 $m = ed - 1$는 n과 서로 소인 모든 a에 대하여 $a^m \equiv 1 \pmod{n}$를 만족한다. 절 3.4.1에서 보았듯이, $\varphi(n)$을 알면 바로 n을 인수분해할 수 있다. 그렇지만 d를 안다고 해서 쉽게 n을 인수분해할 수 있는 것처럼 보이지는 않는다. 그러나 $a^m \equiv 1 \pmod{n}$인 m을 아는 경우에는 "상당히 높은 확률"로 n을 인수분해할 수 있는 확률적인 과정이 있다. (물론 여기에서는 그 확률은 분석하지 않는다.)

알고리듬 3.4.6 (n을 인수분해하는 확률적 알고리듬)**.** 정수 $n = pq$이 서로 다른 두 홀수 소수의 곱이라고 하자. n과 서로 소인 모든 a에 대하여 $a^m \equiv 1 \pmod{n}$을 만족하는 정수 m을 안다고 가정하자. 그러면 이 알고리듬은 "상당히 높은 확률"로 n을 인수분해한다. 아래 각 단계에서 a는 항상 n과 서로 소인 정수를 나타낸다.

1. [2의 거듭제곱들로 나누기] m이 짝수이고 무작위로 선택한 여러 a에 대하여 $a^{m/2} \equiv 1 \pmod{n}$을 만족하면 $m = m/2$으로 교체하여 단계 1로 간다. 그렇지 않은 경우 a는 $a^{m/2} \not\equiv 1 \pmod{n}$이다.
2. [GCD 계산] 무작위로 a를 선택하고 $g = \gcd(a^{m/2} - 1, n)$를 계산한다.
3. [끝?] g가 n의 진약수이면, g를 출력하고 마친다. 그렇지 않으면 Step 2로 간다.

증명을 하기 전에 대수학의 용어를 몇 가지 더 소개한다.

정의 3.4.7 (군 준동형사상)**.** G와 H는 군이다. 함수 $\varphi : G \to H$ 가 **군 준동형사상(group homomorphism)**이라는 것은 모든 $a, b \in G$가 $\varphi(ab) = \varphi(a)\varphi(b)$를 만족하는 것이다. 군 준동형사상이 **전사(surjective)**라고 말할 때는 모든 $c \in H$에 대하여 $\varphi(a) = c$를 만족하는 $a \in G$가 항상 존재하는 경우이다. 군의 준동형사상 $\varphi : G \to H$의 **핵(kernel)**은 $\ker(\varphi)$로 쓰며 $\varphi(a) = 1$를 만족하는 $a \in G$의 집합이다. 군 준동형사상이 $\ker(\varphi) = \{1\}$를 만족하면 **단사(injective)**이다. 참고로 군 준동형사상 $\varphi : G \to H$이 $\ker(\varphi) = \{1\}$이기 위한 필요충분조건은 $\phi(a) = \phi(b)$는 $a = b$를 유도하는 것이다.

정의 3.4.8 (부분군)**.** 만약 G가 군이고 H가 G이 부분집합일 때, H가 G의 연산으로 군이 되면 H를 G의 **부분군(subgroup)**이라고 부른다.

예를 들어, $\varphi : G \to H$가 군준동형사상이면, $\ker(\varphi)$ 는 G의 부분군이다 (Exercise 2.21 참조).

다시 우리의 관심을 알고리듬 3.4.6으로 돌리자. 단계 1에서, $(-1)^m \equiv 1 \pmod{n}$이면 m은 짝수이다. 따라서 $m/2$을 생각하는 것이 의미가 있다. $a^{m/2} \equiv 1 \pmod{n}$이 모든 a에 대해서 성립한다는 것을 확인하는 것은 너무나 많은 시간이 필요하므로 현실적이지 않다. 대신에 몇 개의 난수 a에 대해서만 확인하고, 만약 $a^{m/2} \equiv 1 \pmod{n}$이 확인되면 m을 2로 나눈다. 만약 $a^{m/2} \not\equiv 1 \pmod{n}$인 한 개의 a라도 존재한다면, $a \mapsto a^{m/2}$로 정의된 함수 $(\mathbf{Z}/n\mathbf{Z})^* \to \{\pm 1\}$가 전사 준동형사상이므로 절반의 a도 같은 식을 만족한다.

기초정리 2.5.3으로부터 만약 $x^2 \equiv 1 \pmod{p}$이면 $x = \pm 1 \pmod{p}$임을 안다. Step 2에서, $(a^{m/2})^2 \equiv 1 \pmod{n}$이므로 , $(a^{m/2})^2 \equiv 1 \pmod{p}$와 $(a^{m/2})^2 \equiv 1 \pmod{q}$이 성립하고 따라서 $a^{m/2} \equiv \pm 1 \pmod{p}$과 $a^{m/2} \equiv \pm 1$

$(\mathrm{mod}\ q)$을 얻는다. $a^{m/2} \not\equiv 1\ (\mathrm{mod}\ n)$이므로, 세 가지 가능성이 있고, 다음 두 가능성 중의 하나가 양수의 확률로 나타난다:

1. $a^{m/2} \equiv +1\ (\mathrm{mod}\ p)$ and $a^{m/2} \equiv -1\ (\mathrm{mod}\ q)$
2. $a^{m/2} \equiv -1\ (\mathrm{mod}\ p)$ and $a^{m/2} \equiv +1\ (\mathrm{mod}\ q)$.

유일한 다른 가능성은 모든 부호가 -1인 경우이다. 첫 번째 경우는

$$p \mid a^{m/2} - 1 \qquad \text{이지만} \qquad q \nmid a^{m/2} - 1,$$

이므로 $\gcd(a^{m/2} - 1, pq) = p$가 되고 따라서 n을 인수분해할 수 있다. 마찬가지로 두 번째 경우에는 $\gcd(a^{m/2} - 1, pq) = q$이 성립하므로 역시 n을 인수분해할 수 있다.

예 3.4.9. 다음 매개 변수

$$n = 32295194023343, \qquad e = 29468811804857$$

를 갖는 RSA 암호의 복호화키 $d = 11127763319273$를 알아냈다고 하자. 이 정보와 알고리듬 3.4.6을 이용하여 n을 인수분해한다.

$$m = ed - 1 = 327921963064646896263108960$$

이면 $\varphi(pq) \mid m$이므로 n과 서로 소인 모든 a는 $a^m \equiv 1\ (\mathrm{mod}\ n)$을 만족한다. $a \leq 20$인 모든 a가 $a^{m/2} \equiv 1\ (\mathrm{mod}\ n)$임을 확인한 후 m을

$$\frac{m}{2} = 163960981532323448131554480.$$

으로 바꾼다. 이 새로운 m으로 $a \leq 20$인 모든 a가 $a^{m/2} \equiv 1\ (\mathrm{mod}\ n)$이므로, 또 m을 $\frac{m}{2}$, 즉 819804907661617240657772 40로 바꾼다. 여전히 $a \leq 20$인 모든 a가 $a^{m/2} \equiv 1\ (\mathrm{mod}\ n)$을 만족하므로 다시 m을 409902453830808620328886 20으로 바꾼다. 이제 $2^{m/2} \equiv 4015382800099\ (\mathrm{mod}\ n)$이 되므로 m을 더 이상 2로 나눌 필요는 없다. 이제

$$\gcd(2^{m/2} - 1, n) = \gcd(4015382800098, 32295194023343) = 737531,$$

이므로 n의 약수 737531을 찾았다. 따라서

$$n = 737531 \cdot 43788253.$$

SAGE 예 3.4.10. 알고리듬 3.4.6을 Sage에서 구현한다.

```
sage: def crack_given_decrypt(n, m):
...     n = Integer(n); m = Integer(m);  # some type checking
...     # Step 1: divide out powers of 2
...     while True:
...         if is_odd(m): break
...         divide_out = True
```

```
...         for i in range(5):
...            a = randrange(1,n)
...            if gcd(a,n) == 1:
...               if Mod(a,n)^(m//2) != 1:
...                   divide_out = False
...                   break
...         if divide_out:
...             m = m//2
...         else:
...             break
...     # Step 2: Compute GCD
...     while True:
...         a = randrange(1,n)
...         g = gcd(lift(Mod(a, n)^(m//2)) - 1, n)
...         if g != 1 and g != n:
...             return g
...
```

다음은 위에서 정의한 Sage코드로 예 3.4.9의 계산 결과를 확인하는 방법이다.

```
sage: n=32295194023343; e=29468811804857; d=11127763319273
sage: crack_given_decrypt(n, e*d - 1)
737531
sage: factor(n)
737531 * 43788253
```

훨씬 더 큰 수를 적용하자.

```
sage: e = 22601762315966221465875845336488389513
sage: d = 31940292321834506197902778067109010093
sage: n = 26849492403959099246944675130990465673
sage: p = crack_given_decrypt(n, e*d - 1)
sage: p   # random output (could be other prime divisor)
13432418150982799907
sage: n % p
0
```

3.4.4 참조 추가

RSA암호를 실제로 구현하려고 하면 기억해야 할 요령과 아이디어들이 더 있다. 예를 들어 평문의 각 블록마다 여분의 난수들을 첨가할 수도 있고, 또 어떤 string은 암호화 할 때마다 다르게 암호화할 수 도 있다. 이러한 작업들은 그 전의 암호문과 평문을 알고 있는 공격자라도 그 뒤의 암호문을 공격하기 어렵게 만들 수 있다. 물론 어떤 특정 구현에서는 RSA의 기본구성요소(module)를 인수분해할 필요도 없이 실제로 완벽한 공격이 가능할 수도 있다.

RSA는 OpenSSH protocol version 1 (see `http://www.openssh.com/`)에 사용되는 등 흔히 사용되고 있는 암호 시스템이다. (우리나라에서는 공인인증서에서도 사용되고 있다.)

우리는 절 6.4.2에서 ElGamal 암호을 알아보려고 한다. 이 암호는 RSA와 유사한 느낌을 주지만 어떤 면에서 더 유연하다.

아마도 가장 일반적인 RSA의 공격은 수체 걸르기[1](the number field sieve)인데 이것은 두 소수의 곱 pq 형태의 정수를 인수분해하는 하는 가장 일반적인 알고리듬이다. 이 알고리듬을 설명하는 것은 이 책의 범위를 넘어선다. 일반적으로 사용되는 알고리듬으로 타원 곡선(elliptic curve) 방법이 있는데 절 6.3에서 자세히 설명하려고 한다.

SAGE 예 3.4.11. 수체 걸르기의 변형인 이차 걸르기[2](quadratic sieve)를 이용하여 약 192 비트의 RSA 키를 인수분해하는 간단한 예를 아래에 소개한다.

```
sage: set_random_seed(0)
sage: p = next_prime(randrange(2^96))
sage: q = next_prime(randrange(2^97))
sage: n = p * q
sage: qsieve(n)
([6340271405786663791648052309,
  461023131085921802863987571 59], '')
```

3.5 Exercises

3.1 This problem concerns encoding phrases using numbers using the encoding of Section 3.3.2. What is the longest that an arbitrary sequence of letters (no spaces) can be if it must fit in a number that is less than 10^{20}?

3.2 Suppose Michael creates an RSA cryptosystem with a very large modulus n for which the factorization of n cannot be found in a reasonable amount of time. Suppose that Nikita sends messages to Michael by representing each alphabetic character as an integer between 0 and 26 (`A` corresponds to 1, `B` to 2, etc., and a space ␣ to 0), then encrypts each number *separately* using Michael's RSA cryptosystem. Is this method secure? Explain your answer.

3.3 For any $n \in \mathbf{N}$, let $\sigma(n)$ be the sum of the divisors of n; for example, $\sigma(6) = 1 + 2 + 3 + 6 = 12$ and $\sigma(10) = 1 + 2 + 5 + 10 = 18$. Suppose that $n = pqr$ with p, q, and r distinct primes. Devise an "efficient" algorithm that given n, $\varphi(n)$ and $\sigma(n)$, computes the factorization

[1] sieve가 우리말 체이므로 주로 수체체라 부른다. 소수가 아닌 것을 체로 걸러 낸다는 의미로 걸르기라 번역해 보았음

[2] 이차체

of n. For example, if $n = 105$, then $p = 3$, $q = 5$, and $r = 7$, so the input to the algorithm would be

$$n = 105, \qquad \varphi(n) = 48, \qquad \text{and} \quad \sigma(n) = 192,$$

and the output would be 3, 5, and 7.

3.4 You and Nikita wish to agree on a secret key using the Diffie-Hellman key exchange. Nikita announces that $p = 3793$ and $g = 7$. Nikita secretly chooses a number $n < p$ and tells you that $g^n \equiv 454 \pmod{p}$. You choose the random number $m = 1208$. What is the secret key?

3.5 You see Michael and Nikita agree on a secret key using the Diffie-Hellman key exchange. Michael and Nikita choose $p = 97$ and $g = 5$. Nikita chooses a random number n and tells Michael that $g^n \equiv 3 \pmod{97}$, and Michael chooses a random number m and tells Nikita that $g^m \equiv 7 \pmod{97}$. Brute force crack their code: What is the secret key that Nikita and Michael agree upon? What is n? What is m?

3.6 In this problem, you will "crack" an RSA cryptosystem. What is the secret decoding number d for the RSA cryptosystem with public key $(n, e) = (5352381469067, 4240501142039)$?

3.7 Nikita creates an RSA cryptosystem with public key

$$(n, e) = (1433811615146881, 329222149569169).$$

In the following two problems, show the steps you take to factor n. (Don't simply factor n directly using a computer.)

(a) Somehow you discover that $d = 116439879930113$. Show how to use the probabilistic algorithm of Section 3.4.3 to factor n.

(b) In part (a) you found that the factors p and q of n are very close. Show how to use the Fermat Factorization Method of Section 3.4.2 to factor n.

4
이차상호법칙 (Quadratic Reciprocity)

일차합동식

$$ax \equiv b \pmod{n}$$

가 해를 갖기 위한 필요충분조건은 $\gcd(a, n)$이 b를 나누는 것이다 (기초정리 2.1.15). 이 단원에서는 이차합동식

$$ax^2 + bx + c \equiv 0 \pmod{n}$$

이 해를 가지는지 아닌지를 판단할 수 있는 기준을 찾는 과정에서 찾아낸 아주 놀라운 수학에 대한 이야기를 하려고 한다. 이차 합동식의 풀이는, 대부분의 경우, 중국인의 나머지 정리와 이차방정식의 공식으로부터 어떤 정수 a가 법 p에 관하여 완전제곱수인가의 문제로 귀결된다.

Gauss의 이차상호법칙은

정수 a가 $(\mathbf{Z}/p\mathbf{Z})^*$에서 완전제곱수가 되는 p는 어떤 소수들인가?

에 대한 정확한 답을 제공한다. 이 단원에서 우리가 완벽하게 증명할 심오한 사실은 그 답이 $p \pmod{4a}$에 의존한다는 것이다. 따라서 *a 가 법 p에 관하여 제곱수라면, p를 $4a$로 나눈 나머지만을 고려하면 된다*는 것은 굉장히 놀라운 발견이다. 또한 이차상호법칙은 class field theory와 Langlands program과 같은 현대 고등 정수론 연구의 중심이 되고 있다.

이차상호법칙의 증명은 100가지가 넘는다(긴 목록은 [31]을 참조). 이 단원에서는 우리는 두 가지 증명을 소개한다. 절 4.3에서 소개하는 첫 번째 증명은 아주 기본적인 증명으로 어떤 구간에서의 정수인 점들을 추적하는 방법을 포함하고 있다. 이 증명은 추상적인 개념을 많이 쓰지 않고 모든 세부적인 증명을 이해할 수 있는 장점이 있지만, 무엇을 하고 있는지가 개념적으로 잘 이해가

안 되는 경우에는 만족스럽지 못 할 수도 있다. 반면에, 4.4절에서 증명 할 두 번째 증명은 좀 더 추상적이고 Gauss 합(sum)에 관한 개념과 성질들을 사용한다. 이 절을 공부하는 독자들은 따라서 4.1절과 4.2절을 꼭 읽은 후 각자의 취향과 또 추상대수학을 어느 정도 공부했는지에 따라 4.3절이나 4.4절 중 한 절을 공부하면 된다.

4.5절에서는 다시 제곱근의 계산과 이차합동식을 푸는 실질적인 계산법을 다룬다.

4.1 이차상호법칙

이차상호법칙을 이 절에서 기술한다.

정의 4.1.1 (이차잉여)**.** 소수 p를 고정하자. p로 나누어지지 않는 정수 a가 법 p에 관하여 어떤 수의 제곱과 같아지면 a를 법 p의 **이차잉여(quadratic residue)**라고 하고, 그렇지 않으면 a는 **이차비잉여(quadratic nonresidue)**라고 한다.

예를 들어 법 5에 관하여

$$1^2 = 1, \quad 2^2 = 4, \quad 3^2 = 4, \quad 4^2 = 1, \qquad (\mathrm{mod}\ 5)$$

이므로, 1과 4는 이차잉여이고, 2와 3은 이차비잉여이다.

이차상호법칙 정리는 이 책에서 증명하려고 하는 가장 심오한 정리이다. 이 정리는 정수 a가 법 p의 이차잉여인가에 대한 문제를 a의 각 소수 약수가 법 p의 이차잉여인가의 문제와 연결시킨다. 이를 정확하게 기술하기 위하여 새로운 표기법을 소개한다.

정의 4.1.2 (Legendre 부호)**.** p는 홀수 소수이고 a는 정수이다.

$$\left(\frac{a}{p}\right) = \begin{cases} 0 & (\gcd(a,p) \neq 1\text{이면}) \\ +1 & (a\text{가 이차잉여이면}) \\ -1 & (a\text{가 이차비잉여이면}) \end{cases}$$

라고 약속하고 이 부호를 **Legendre 부호(Legendre Symbol)**라고 부른다.

예를 들어

$$\left(\frac{1}{5}\right) = 1, \quad \left(\frac{2}{5}\right) = -1, \quad \left(\frac{3}{5}\right) = -1, \quad \left(\frac{4}{5}\right) = 1, \quad \left(\frac{5}{5}\right) = 0$$

이 성립한다.

SAGE 예 4.1.3. Sage에서는 `legendre_symbol`를 사용하여 Legendre 부호를 계산한다.

```
sage: legendre_symbol(2,3)
-1
sage: legendre_symbol(1,3)
1
sage: legendre_symbol(3,5)
-1
sage: legendre_symbol(Mod(3,5), 5)
-1
```

$\left(\frac{a}{p}\right)$는 오직 $a \pmod p$에만 의존하므로, $a \in \mathbf{Z}/p\mathbf{Z}$에 대하여 $\left(\frac{a}{p}\right)$를 a의 임의의 올림 $\tilde{a}$를 잡아 $\left(\frac{\tilde{a}}{p}\right)$로 정의하여도 같은 값이어야 한다.

군 준동형사상 (정의 3.4.7참조) $\varphi : G \to H$는 임의의 $a, b \in G$가 $\varphi(ab) = \varphi(a)\varphi(b)$를 만족하는 함수이다. 또 임의의 $c \in H$에 대하여 $\varphi(a) = c$를 만족하는 $a \in G$가 존재하면 φ는 전사함수이다. 다음 보조정리는 이차잉여부호가 어떻게 전사인 준동형사상을 정의하는지를 보여준다.

보조정리 4.1.4. *함수 $\psi : (\mathbf{Z}/p\mathbf{Z})^* \to \{\pm 1\}$를 $\psi(a) = \left(\frac{a}{p}\right)$로 정의하면 ψ는 전사인 준동형사상이다. 달리 표현하면 ϕ는 곱셈적 함수이다.*

증명 정리 2.5.8로부터 소수 p의 원시근 g이 존재한다. 따라서

$$g, g^2, \ldots, g^{(p-1)/2}, g^{(p+1)/2}, \ldots, g^{p-1} = 1$$

이 $(\mathbf{Z}/p\mathbf{Z})^*$의 모든 원소들이다. $p-1$이 짝수이므로, $(\mathbf{Z}/p\mathbf{Z})^*$의 원소들의 제곱은

$$g^2, g^4, \ldots, g^{(p-1)/2\cdot 2} = 1, g^{p+1} = g^2, \ldots, g^{2(p-1)} = 1$$

이다. $(\mathbf{Z}/p\mathbf{Z})^*$의 원소들의 제곱들의 리스트에서 $(g^{(p+1)/2})^2 = g^{p+1} = g^2$부터 시작하여 그 뒤를 이어 나타나는 모든 g의 거듭제곱들은 그 전에 나타난 것들이다. 따라서 $(\mathbf{Z}/p\mathbf{Z})^*$에서의 제곱수들은 정확히 $g^n,\ n = 2, 4, \ldots, p-1$이고 $g^n,\ n = 1, 3, \ldots, p-2$는 제곱수가 아니다. 그런데 홀수 더하기 홀수는 짝수, 짝수 더하기 짝수는 짝수, 홀수 더하기 짝수는 홀수이므로 ψ는 준동형사상이다. 게다가 g가 제곱수가 아니므로 $\psi(g) = -1$이고, 따라서 ψ는 전사함수이다.

□

참조 4.1.5. [1] 위의 증명을 군론의 언어로 바꿔보자. 군 $G = (\mathbf{Z}/p\mathbf{Z})^*$는 위수가 $p-1$인 순환군이다. p가 홀수이므로 $p-1$은 짝수이고 G의 제곱수들로 이루어진 G의 부분군 H의 지표(index)는 2이다. (H가 부분군인 이유는 Exercise 4.2를 참조하기 바란다.) $\left(\frac{a}{p}\right) = 1$이라는 것은 $a \in H$와 동치이므로, ψ는 두 함수 $G \to G/H \cong \{\pm 1\}$의 합성이다. 단, 두 번째 동형함수는 G/H의 H는 1에, 나머지 원소는 -1과 대응시킨 함수이다.

[1] 군론의 지식이 있는 경우 참조-역주

TABLE 4.1. 어떤 소수 p의 법에서 5가 제곱수인가?

p	$\left(\frac{5}{p}\right)$	$p \bmod 5$	p	$\left(\frac{5}{p}\right)$	$p \bmod 5$
7	-1	2	29	1	4
11	1	1	31	1	1
13	-1	3	37	-1	2
17	-1	2	41	1	1
19	1	4	43	-1	3
23	-1	3	47	-1	2

참조 4.1.6. $(\mathbf{Z}/p\mathbf{Z})^*$가 순한군이라는 사실을 이용하지 않고 다음과 같이 다른 방법으로 증명할 수 있다. 만약 $a \in (\mathbf{Z}/p\mathbf{Z})^*$가 제곱수라면 $a \equiv b^2 \pmod p$라고 쓸 수 있고, 그러면 $a^{(p-1)/2} = b^{p-1} \equiv 1 \pmod p$이므로, a는 $f = x^{(p-1)/2} - 1$의 해이다. 기초정리 2.5.3에 의하여, 다항식 f는 많아야 $(p-1)/2$개의 해를 갖는다. 따라서 f의 해가 되지 않는 $(\mathbf{Z}/p\mathbf{Z})^*$의 원소 a가 존재한다. 이 a는 $\psi(a) = \left(\frac{a}{p}\right) = -1$를 만족하고, 또 자명하게 $\psi(1) = 1$이므로, ψ는 전사함수이다. 물론 이 설명으로는 ψ가 준동형사상임은 보일 수 없다.

부호 $\left(\frac{a}{p}\right)$는 a의 법 p의 잉여류에만 의존한다. 그래서 가능한 모든 a에 대하여 $\left(\frac{a}{5}\right)$의 표를 만드는 건 쉽다. 많은 소수 p에 대하여 $\left(\frac{5}{p}\right)$에 대한 표를 만드는 것도 쉬울까? 표 4.1을 보면 아마도 어떤 간단한 패턴이 있는 것처럼 보인다. $\left(\frac{5}{p}\right)$가 $p \pmod 5$에 의존하는 것처럼 보인다. 좀 더 자세히 설명하면 $\left(\frac{5}{p}\right) = 1$이기 위한 필요충분조건이 $p \equiv 1, 4 \pmod 5$인 것 처럼 보인다. 즉, $\left(\frac{5}{p}\right) = 1$이기 위한 필요충분조건이 p가 법 5의 제곱수인 것이다.

비슷한 관찰에 근거하여, 18세기에 여러 수학자들이 표 4.1가 제안하는 신비로운 현상에 대한 추론적인 설명들을 찾아내었다. 드디어 1796년 4월 8일 19세의 Gauss가 다음 정리를 증명하였다.

정리 4.1.7 (Gauss의 이차상호법칙)**.** *p와 q는 서로 다른 홀수 소수이다. 그러면*

$$\left(\frac{p}{q}\right) = (-1)^{\frac{p-1}{2}\cdot\frac{q-1}{2}} \left(\frac{q}{p}\right)$$

이 성립한다. 또

$$\left(\frac{-1}{p}\right) = (-1)^{(p-1)/2} \qquad \text{와} \qquad \left(\frac{2}{p}\right) = \begin{cases} 1 & \text{if } p \equiv \pm 1 \pmod 8 \\ -1 & \text{if } p \equiv \pm 3 \pmod 8 \end{cases}$$

도 성립한다.

$\left(\frac{p}{q}\right)$와 $\left(\frac{q}{p}\right)$를 연결하는 이 Gauss 공식의 증명을 두 가지 소개한다. 첫 번째 초보적인 증명은 4.3절에서, 좀 더 대수적인 두 번째 증명은 4.4절에서 한다.

우리의 예에 가우스 정리를 적용하면

$$\left(\frac{5}{p}\right) = (-1)^{2\cdot\frac{p-1}{2}}\left(\frac{p}{5}\right) = \left(\frac{p}{5}\right) = \begin{cases} +1 & \text{if } p \equiv 1,4 \pmod 5 \\ -1 & \text{if } p \equiv 2,3 \pmod 5. \end{cases}$$

응용으로서, 다음 예는 법 b의 나머지로서 a가 제곱수인가 하는 질문에 정리 4.1.7을 어떻게 적용하는 지를 보여준다.

예 4.1.8. 389를 법으로 할 때 69는 제곱수인가? 여기서 389는 소수이다.

$$\left(\frac{69}{389}\right) = \left(\frac{3\cdot 23}{389}\right) = \left(\frac{3}{389}\right)\cdot\left(\frac{23}{389}\right)$$

인데,

$$\left(\frac{3}{389}\right) = \left(\frac{389}{3}\right) = \left(\frac{2}{3}\right) = -1,$$

$$\begin{aligned}\left(\frac{23}{389}\right) &= \left(\frac{389}{23}\right) = \left(\frac{21}{23}\right) = \left(\frac{-2}{23}\right) \\ &= \left(\frac{-1}{23}\right)\left(\frac{2}{23}\right) = (-1)^{\frac{23-1}{2}}\cdot 1 = -1\end{aligned}$$

이므로,

$$\left(\frac{69}{389}\right) = (-1)(-1) = 1$$

이다. 따라서 69는 법 389의 나머지로써(간단히 법 389에서) 제곱수이다.

SAGE 예 4.1.9. Sage에서의 이 계산은 다음과 같다.

```
sage: legendre_symbol(69,389)
1
```

69는 법 389에서 제곱수이지만, $x^2 \equiv 69 \pmod{389}$을 만족하는 x는 아직 알지 못한다. 이는 인수분해를 하지도 않고 어떤 수가 합성수임을 증명했던 정리 2.4.1를 연상시킨다.

참조 4.1.10. 소수 법에서의 Legendre 부호를 합성수 법까지 확장한 것을 Jacobi 부호라고 한다. 자세한 것은 Exercise 4.9을 참조하기 바란다.

4.2 Euler의 기준

p는 홀수 소수이고 a는 p로 나누어지지 않는 정수이다. Euler는 $\left(\frac{a}{p}\right)$가 $a^{(p-1)/2}$와 법 p에서 합동인 사실을 증명할 때 원시근의 존재를 이용하였다. 정리 4.1.7의 두 가지 증명 모두에 이 사실을 반복적으로 사용한다.

기초정리 4.2.1 (오일러의 기준 (Euler's Criterion))**.**

$$\left(\frac{a}{p}\right) = 1 \qquad \Longleftrightarrow \qquad a^{(p-1)/2} \equiv 1 \pmod{p}.$$

증명 함수 $\varphi : (\mathbf{Z}/p\mathbf{Z})^* \to (\mathbf{Z}/p\mathbf{Z})^*$를 $\varphi(a) = a^{(p-1)/2}$ 로 정의하자. 그러면 쉽게 군의 준동형사상임을 보일 수 있다(Exercise 4.2). 함수 $\psi : (\mathbf{Z}/p\mathbf{Z})^* \to \{\pm 1\}$는 $\psi(a) = \left(\frac{a}{p}\right)$인 보조정리 4.1.4의 준동형사상이다. 만약 $a \in \ker(\psi)$ 이면, 적당한 $b \in (\mathbf{Z}/p\mathbf{Z})^*$가 존재하여 $a = b^2$을 만족하므로,

$$\varphi(a) = a^{(p-1)/2} = (b^2)^{(p-1)/2} = b^{p-1} = 1$$

이 성립한다. 따라서, $\ker(\psi) \subset \ker(\varphi)$이다. 보조정리 4.1.4로부터, $\ker(\psi)$는 $(\mathbf{Z}/p\mathbf{Z})^*$에서 index가 2이다. 즉, $\#(\mathbf{Z}/p\mathbf{Z})^* = 2 \cdot \#\ker(\psi)$이다. 준동형사상의 핵도 군이고 부분군의 위수는 군의 위수를 나누므로, $\ker(\varphi) = \ker(\psi)$ 거나 $\varphi = 1$의 두 경우가 가능하다. 만약 $\varphi = 1$이면, 다항식 $x^{(p-1)/2} - 1$은 체 $\mathbf{Z}/p\mathbf{Z}$에 $p-1$개의 해를 가지므로 기초정리 2.5.3에 모순이다. 그러므로 $\ker(\varphi) = \ker(\psi)$가 같아진다. 따라서 이 기초정리가 성립한다. □

SAGE 예 4.2.2. 따름정리 4.2.3로부터 $\left(\frac{a}{p}\right)$를 편리하게 계산할 수 있다. 이를 Sage에서 예시한다.

```
sage: def kr(a, p):
...     if Mod(a,p)^((p-1)//2) == 1:
...        return 1
...     else:
...        return -1
sage: for a in range(1,5):
...     print a, kr(a,5)
1 1
2 -1
3 -1
4 1
```

따름정리 4.2.3. *합동방정식 $x^2 \equiv a \pmod{p}$가 해를 갖지 않을 필요충분조건은 $a^{(p-1)/2} \equiv -1 \pmod{p}$이다. 따라서 $\left(\frac{a}{p}\right) \equiv a^{(p-1)/2} \pmod{p}$가 항상 성립한다.*

증명 기초정리 4.2.1와 다항식 $x^2 - 1$은 $+1$, -1만 해로 가진다는 사실(기초정리 2.5.5로 부터 유도되는)을 적용한다. □

만약 a와 p가 둘 다 매우 큰 수가 되면, 정리 4.1.7를 활용한 $\left(\frac{a}{p}\right)$의 계산은 a의 소인수분해를 요구하므로 실용적이지는 못하다. 따라서 따름정리 4.2.3에 나오는 식을 계산하기 위한 추가적인 방법을 탐구할 필요가 있다.

예 4.2.4. $p = 11$이라고 가정하자. $(\mathbf{Z}/11\mathbf{Z})^*$의 모든 원소를 제곱함으로써, 법 11에서의 제곱들은 $\{1, 3, 4, 5, 9\}$임을 안다. 다음은 $(\mathbf{Z}/11\mathbf{Z})^*$의 모든 원소 a에 대하여 $a^{(p-1)/2} = a^5$를 계산한 결과이다.

$$1^5 = 1,\ 2^5 = -1,\ 3^5 = 1,\ 4^5 = 1,\ 5^5 = 1,$$
$$6^5 = -1,\ 7^5 = -1,\ 8^5 = -1,\ 9^5 = 1,\ 10^5 = -1.$$

계산으로부터 $a^5 = 1$이 되는 a는 $\{1, 3, 4, 5, 9\}$인데, 기초정리4.2.1에서 예상한 결과이다.

예 4.2.5. 소수 $p = 726377359$를 법으로 할 때 3이 제곱수인지 아닌지를 결정하자.

```
sage: p = 726377359
sage: Mod(3, p)^((p-1)//2)
726377358
```

이므로

$$3^{(p-1)/2} \equiv -1 \pmod{726377359}$$

이다. 따라서 3은 법이 p일 때 제곱수가 아니다. 이 계산은 어렵지는 않지만 손으로 한다면 굉장히 지겨운 일일것이다. 3은 작은 수이므로 이차잉여법칙을 이용하면 아주 간단한 손계산으로도 답을 얻을 수 있다.

$$\left(\frac{3}{726377359}\right) = (-1)^{((3-1)/2)\cdot((726377359-1)/2)} \left(\frac{726377359}{3}\right)$$
$$= (-1) \cdot \left(\frac{1}{3}\right) = -1.$$

4.3 이차상호법칙의 첫 번째 증명

이차상호법칙의 첫 번째 증명은 구간에서의 정수들을 추적하는 것을 포함하는 등 초보적이다. 그 첫 번째 단계로 어떤 구간에 포함된 특별한 형태의 정수들의 개수로 $\left(\frac{a}{p}\right)$를 계산하는 가우스의 보조정리를 증명한다. 그 다음 구간의 끝 점들이 바뀔 때 각 구간안의 정수점들의 개수의 비율을 제어하는 방법을 보여주는 보조정리 4.3.3를 증명한다. 그런 후 구간의 끝 점이 바뀔 때마마 그 구간들의 정수점들의 개수가 어떻게 변하는 지를 조심스럽게 추적하고 가우스의 보조정리를 적용하여, $\left(\frac{a}{p}\right)$는 $p \pmod{4a}$에 의존하는 것을 보인다. 마지막으로 간단한 대수를 사용하여 우리가 막 찾아 낸 도구들로부터 이차상호법칙을 끌어낸다. 이 증명은 [15]에 있는 증명을 참조하였다.

보조정리 4.3.1 (Gauss의 보조정리)**.** *p는 홀수소수이고 정수 a는 $a \not\equiv 0 \pmod{p}$를 만족한다. $\frac{p-1}{2}$개의 정수*

$$a,\ 2a,\ 3a,\ \ldots,\ \frac{p-1}{2}a$$

를 각각 법 p에 관하여 구간 $(-\frac{p}{2},\ \frac{p}{2})$에 합동인 점과 대응시켰을 때 음수가 되는 수들의 개수를 ν라고 하자. 그러면

$$\left(\frac{a}{p}\right) = (-1)^{\nu}$$

가 성립한다.

증명 집합

$$S = \left\{a, 2a, \ldots, \frac{p-1}{2}a\right\}$$

에 있는 수들은 법 p에서 모두 다르다. 따라서 S의 원소를 법 p로 하여 구간 $(-\frac{p}{2},\ \frac{p}{2})$에 속하도록 계산을 하여 얻은 집합을 T라고 하면 집합 T의 원소의 개수는 $(p-1)/2$ 개다. 또 S의 임의의 두 수의 합도 p의 배수가 아니다. 왜냐하면 $1 \le k, j \le (p-1)/2$에 대하여 $ka + ja \equiv 0 \pmod{p}$이라면 $k + j \equiv 0 \pmod{p}$가 되는데 이는 모순이다. 그러므로

$$\left\{1, -1, 2, -2, \ldots, \frac{p-1}{2}, -\frac{p-1}{2}\right\}$$

의 부분집합인 T에는 j와 $-j$가 동시에 속하지는 않는다. 따라서 $k \in \{1, 2, \ldots, \frac{p-1}{2}\}$이면 k와 $-k$중 정확히 하나만 T의 원소여야만 한다. 즉,

$$T = \left\{\varepsilon_1 \cdot 1, \varepsilon_2 \cdot 2, \ldots, \varepsilon_{(p-1)/2} \cdot \frac{p-1}{2}\right\}.$$

이 때 ε_i는 $+1$ 혹은 -1중 하나이고 ν는 $\varepsilon_i = -1$인 i들의 개수이다. S의 모든 원소들의 곱과 T의 모든 원소들의 곱은 법 p에서 합동이므로

$$(1a) \cdot (2a) \cdot (3a) \cdots \left(\frac{p-1}{2}a\right) \equiv (\varepsilon_1 \cdot 1) \cdot (\varepsilon_2 \cdot 2) \cdots \left(\varepsilon_{(p-1)/2} \cdot \frac{p-1}{2}\right) \pmod{p}$$

을 얻는다. 따라서

$$a^{(p-1)/2} \equiv \varepsilon_1 \cdot \varepsilon_2 \cdots \varepsilon_{(p-1)/2} \equiv (-1)^{\nu} \pmod{p}.$$

기초정리 4.2.1로부터 $\left(\frac{a}{p}\right) = a^{(p-1)/2}$이므로 우리의 보조정리가 증명된다. □

SAGE 예 4.3.2. Sage를 이용하여 가우스의 보조정리를 구체적인 예에서 확인한다. 아래의 `gauss` 함수는 가우스의 보조정리의 증명에서 엄급한 집합 T의 원소들을 출력한다. 아래의 각 경우에서, $(-1)^{\nu} = \left(\frac{a}{p}\right)$이다.

```
sage: def gauss(a, p):
...    # make the list of numbers reduced modulo p
...    v = [(n*a)%p for n in range(1, (p-1)//2 + 1)]
...    # normalize them to be in the range -p/2 to p/2
...    v = [(x if (x < p/2) else x - p) for x in v]
...    # sort and print the resulting numbers
...    v.sort()
...    print v
...    # count the number that are negative
...    num_neg = len([x for x in v if x < 0])
...    return (-1)^num_neg
sage: gauss(2, 13)
[-5, -3, -1, 2, 4, 6]
-1
sage: legendre_symbol(2,13)
-1
sage: gauss(4, 13)
[-6, -5, -2, -1, 3, 4]
1
sage: legendre_symbol(4,13)
1
sage: gauss(2,31)
[-15, -13, -11, -9, -7, -5, -3, -1, 2, 4, 6, 8, 10, 12, 14]
1
sage: legendre_symbol(2,31)
1
```

4.3.1 Euler의 기초정리

유리수 $a, b \in \mathbf{Q}$에 대하여,

$$(a,b) \cap \mathbf{Z} = \{x \in \mathbf{Z} : a < x < b\}$$

는 a와 b사이의 정수들의 집합이다. 다음 보조정리는 어떤 열린 구간에 있은 정수들의 개수를 어떻게 추적하는지를 보여준다.

보조정리 4.3.3. *a, b는 유리수이다. 그러면 임의의 정수 n에 대하여, 합동식에 나타나는 각 구간이 공집합이 아니라면, 다음이 성립한다.*

$$\#\left((a,b) \cap \mathbf{Z}\right) \equiv \#\left((a, b+2n) \cap \mathbf{Z}\right) \pmod 2,$$

$$\#\left((a,b) \cap \mathbf{Z}\right) \equiv \#\left((a-2n, b) \cap \mathbf{Z}\right) \pmod 2.$$

만약 고려해야하는 구간 들 중 하나라도 공집합이면 위의 명제는 참이 아닐 수 있다. 예를 들어, $(a,b) = (-1/2, 1/2)$이고 $n = -1$이면, $\#((a,b) \cap \mathbf{Z}) = 1$이나 $\#(a, b-2) \cap \mathbf{Z} = 0$이다.

증명 $\lceil x \rceil$는 x보다 크거나 같은 정수 중 가장 작은 정수라고 하자. $n > 0$이라면

$$(a, b + 2n) = (a, b) \cup [b, b + 2n),$$

이고 (a, b)와 $[b, b + 2n)$는 공통부분이 없다. 그런데 구간 $[b, b + 2n)$에 놓인 정수는

$$\lceil b \rceil, \lceil b \rceil + 1, \ldots, \lceil b \rceil + 2n - 1$$

으로 $2n$ 개이다. 따라서 $n > 0$인 경우 첫 번째 합동식이 성립한다. 또

$$(a, b - 2n) = (a, b) \text{ minus } [b - 2n, b)$$

이고 $[b - 2n, b)$도 정확히 $2n$ 개의 정수들을 포함하므로 n이 음수인 경우에도 첫 번째 합동식은 성립한다. $\#\left((a - 2n, b) \cap \mathbf{Z}\right)$에 관한 두 번째 합동식도 비슷하게 증명할 수 있다. □

일단 다음 기초정리를 증명하면, 이차상호법칙을 추론하는 것은 어렵지 않다.

기초정리 4.3.4 (Euler)**.** *p는 홀수 소수이고 a는 $p \nmid a$인 양의 정수이다. 정수 q는 $q \equiv \pm p \pmod{4a}$인 소수라면 $\left(\frac{a}{p}\right) = \left(\frac{a}{q}\right)$이 성립한다.*

증명 보조정리를 4.3.1를 적용하여 $\left(\frac{a}{p}\right)$를 계산한다.

$$S = \left\{a, 2a, 3a, \ldots, \frac{p-1}{2}a\right\}$$

라 놓고,

$$I = \left(\frac{1}{2}p, p\right) \cup \left(\frac{3}{2}p, 2p\right) \cup \cdots \cup \left(\left(b - \frac{1}{2}\right)p, bp\right),$$

라 하자. 단, $b = \frac{1}{2}a$, 혹은 $\frac{1}{2}(a-1)$로 정수값이다.

S의 원소 중 법 p에서 구간 $(-\frac{p}{2}, 0)$에 있는 원소와 합동이 되는 원소들은 I에 속하고 있음을 확인하자. 먼저 $b = \frac{1}{2}a$인 경우는

$$bp = \frac{1}{2}ap = \frac{p}{2}a > \frac{p-1}{2}a$$

이므로 S의 모든 원소는 I에 속한다. 다음 $b = \frac{1}{2}(a-1)$인 경우는

$$bp + \frac{p}{2} = \frac{a-1}{2}p + \frac{p}{2} = \frac{p-1+a}{2} > \frac{p-1}{2}a$$

이므로 $((b - \frac{1}{2})p, bp)$가 S의 원소를 포함할 수 있는 마지막 구간이다. 참고로 I의 각 구간의 끝점은 p의 배수이거나 정수가 아니므로 S에 속하지 않는다. 이제 보조정리 4.3.1를 적용하면

$$\left(\frac{a}{p}\right) = (-1)^{\#(S \cap I)}$$

를 얻는다.

먼저 $\#(S\cap I)$를 계산하기 위해서는 S와 I의 각 구간을 a로 나누어 정수점을 찾으면 되는 것을 관찰하자. 즉,

$$\#(S\cap I) = \#\left(\frac{1}{a}S\cap\frac{1}{a}I\right) = \#\left(\mathbf{Z}\cap\frac{1}{a}I\right).$$

위 식에서

$$\frac{1}{a}I = \left(\left(\frac{p}{2a},\frac{p}{a}\right)\cup\left(\frac{3p}{2a},\frac{2p}{a}\right)\cup\cdots\cup\left(\frac{(2b-1)p}{2a},\frac{bp}{a}\right)\right),$$

$\frac{1}{a}S = \{1,2,3,4,\ldots,(p-1)/2\}$이다. 또 두 번 째 등식은 $\frac{1}{a}I\subset(0,(p-1)/2+1/2]$이기 대문인데, 이 사실은

$$\frac{pb}{a}\le\frac{p\frac{a}{2}}{a} = \frac{p}{2} = \frac{p-1}{2}+\frac{1}{2}$$

으로부터 알 수 있다.

$p=4ac+r$이라 놓고

$$J = \left(\left(\frac{r}{2a},\frac{r}{a}\right)\cup\left(\frac{3r}{2a},\frac{2r}{a}\right)\cup\cdots\cup\left(\frac{(2b-1)r}{2a},\frac{br}{a}\right)\right)$$

라 하자. $\frac{1}{a}I$와 J의 유일한 다른 점은, 예를 들어

$$\frac{r}{2a}-\frac{p}{2a} = \frac{p}{2a}-2c-\frac{p}{2a} = -2c.$$

인 것처럼, 구간의 각 끝 점들이 2의 배수만큼 바뀐 것이다. 보조정리 4.3.3로부터

$$\nu = \#\left(\mathbf{Z}\cap\frac{1}{a}I\right)\equiv\#(\mathbf{Z}\cap J)\pmod 2.$$

따라서 $\left(\frac{a}{p}\right) = (-1)^\nu$는 r과 a에만 의존한다. 좀 더 정확하게 $p \pmod{4a}$에 의존한다. 따라서 $q\equiv p \pmod{4a}$이면 $\left(\frac{a}{p}\right) = \left(\frac{a}{q}\right)$이다.

만약 $q\equiv -p \pmod{4a}$이면 위의 계산에서 r을 $4a-r$로 바꾸면 된다. 그러면 J는 다음 집합 K로 바뀐다.

$$K = \left(2-\frac{r}{2a},4-\frac{r}{a}\right)\cup\left(6-\frac{3r}{2a},8-\frac{2r}{a}\right)\cup\cdots \cup\left(4b-2-\frac{(2b-1)r}{2a},4b-\frac{br}{a}\right).$$

그러므로 K는 끝점이 짝수 배만큼 더해진 점을 제외하고는 $-J$와 같다. 보조정리 4.3.3로부터

$$\#(K\cap\mathbf{Z})\equiv\#\left(\frac{1}{a}I\cap\mathbf{Z}\right)\pmod 2$$

를 얻으므로, 이번에도 $\left(\frac{a}{p}\right) = \left(\frac{a}{q}\right)$이 성립한다. □

$a = 2$이면 집합을 조심스럽게 관찰함으로써 모든 가능한 경우에 보조정리 4.3.1의 ν를 구할 수 있으며, 그 결과는 종종 아주 유용하게 쓰인다. 아래 기초정리의 다른 증명은 Exercise 4.6를 참조한다.

기초정리 4.3.5 (Legendre Symbol of 2)**.** *p는 홀수 소수이다. 그러면*

$$\left(\frac{2}{p}\right) = \begin{cases} 1 & \text{if } p \equiv \pm 1 \pmod 8 \\ -1 & \text{if } p \equiv \pm 3 \pmod 8 \end{cases}$$

이 성립한다.

증명 $a = 2$이면 집합 $S = \{a, 2a, \ldots, \cdot\frac{p-1}{2}a\}$는

$$\{2, 4, 6, \ldots, p-1\}$$

이다. 이제 S의 원소들 중 법 p에서 구간 $I = (\frac{p}{2}, p)$에 들어 가는 원소들의 개수를 세어야 한다. $p = 8c + r$로 놓으면,

$$\begin{aligned} \#(I \cap S) &= \#\left(\frac{1}{2}I \cap \mathbf{Z}\right) = \#\left(\left(\frac{p}{4}, \frac{p}{2}\right) \cap \mathbf{Z}\right) \\ &= \#\left(\left(2c + \frac{r}{4}, 4c + \frac{r}{2}\right) \cap \mathbf{Z}\right) \equiv \#\left(\left(\frac{r}{4}, \frac{r}{2}\right) \cap \mathbf{Z}\right) \pmod 2 \end{aligned}$$

를 얻고, 위 식에서 마지막 등호는 보조정리 4.3.3덕이다. 이제 r의 가능성은 $1, 3, 5, 7$뿐이다. $r = 1$이면, 위 집합의 개수는 0, $r = 3, 5$이면 1, $r = 7$이면 2이므로 증명이 완성된다. □

4.3.2 이차잉여 법칙의 증명

이제 간단히 이차잉여 법칙을 증명할 수 있다.

정리 4.1.7의 첫 번째 증명 먼저 $p \equiv q \pmod 4$라고 가정하자. 필요하다면 p와 q를 교환하여 $p > q$라고 가정하고, $p - q = 4a$로 표현하자. $p = 4a + q$이므로

$$\left(\frac{p}{q}\right) = \left(\frac{4a+q}{q}\right) = \left(\frac{4a}{q}\right) = \left(\frac{4}{q}\right)\left(\frac{a}{q}\right) = \left(\frac{a}{q}\right)$$

이고

$$\left(\frac{q}{p}\right) = \left(\frac{p-4a}{p}\right) = \left(\frac{-4a}{p}\right) = \left(\frac{-1}{p}\right) \cdot \left(\frac{a}{p}\right)$$

이다. $p \equiv q \pmod{4a}$이므로 기초정리 4.3.4에 의하여 $\left(\frac{a}{q}\right) = \left(\frac{a}{p}\right)$이다. 따라서

$$\left(\frac{p}{q}\right) \cdot \left(\frac{q}{p}\right) = \left(\frac{-1}{p}\right) = (-1)^{\frac{p-1}{2}} = (-1)^{\frac{p-1}{2} \cdot \frac{q-1}{2}}$$

을 얻는데, 마지막 등호는 $p - q = 4a$인 경우는 $\frac{p-1}{2}$이 짝수라는 사실과 $\frac{q-1}{2}$이 짝수라는 사실이 같기 때문이다.

이제 $p \not\equiv q \pmod 4$라고 하자. 그러면 $p \equiv -q \pmod 4$이므로 $p+q=4a$로 쓸 수 있다. 그러면 다음이 성립한다.

$$\left(\frac{p}{q}\right) = \left(\frac{4a-q}{q}\right) = \left(\frac{a}{q}\right), \qquad \left(\frac{q}{p}\right) = \left(\frac{4a-p}{p}\right) = \left(\frac{a}{p}\right).$$

또 기초정리 4.3.4로부터 $p \equiv -q \pmod 4$는 $\left(\frac{a}{q}\right) = \left(\frac{a}{p}\right)$를 유도한다. 따라서 $\left(\frac{p}{q}\right)\left(\frac{q}{p}\right) = \left(\frac{a}{q}\right)\left(\frac{a}{p}\right) = \left(\frac{a}{q}\right)^2 = 1$이다. 또 $p \equiv -q \pmod 4$는 $\frac{p-1}{2}$와 $\frac{q-1}{2}$ 둘 중 하나는 짝수이고 나머지는 홀수임을 유도하므로 $(-1)^{\frac{p-1}{2}\cdot\frac{q-1}{2}} = 1$이 되어 증명이 완성된다. □

4.4 가우스 합을 이용한 이차상호 법칙의 증명

이 절에서는 "1의 근"들의 합들이 만족하는 대수적 등식을 이용하여 정리 4.1.7의 멋진 증명을 소개한다. 이 증명에서 소개하는 대상들은 그 자체로도 충분히 흥미롭고 더 높은 차수에서의 이차상호와 유사한 법칙을 증명할 때 강력한 도구가 된다. (더 높은 차수의 상호법칙은 아래 증명의 모델인 [25]의 6절을 참고한다.)

정의 4.4.1 (단위원의 해)**.** n 번째 **단위원의 해(roots of unity)**는 $\zeta^n = 1$를 만족하는 복소수 ζ를 말한다. 단위원의 해 ζ를 단위원의 n 번째 **원시근(primitive root)**이라고 부를 때는 n이 $\zeta^n = 1$을 만족하는 가장 작은 양의 정수가 될 때이다.

예를 들어, -1는 단위원의 두 번째 원시근이고, $\zeta = \frac{\sqrt{-3}-1}{2}$는 단위원의 세 번째 원시근이다. 일반적으로, 임의의 $n \in \mathbf{N}$에 대하여, 복소수

$$\zeta_n = \cos(2\pi/n) + i\sin(2\pi/n)$$

는 단위원의 n 번째 원시근이다 (이 사실은 $e^{i\theta} = \cos(\theta) + i\sin(\theta)$로부터 따라온다). 이 절의 나머지에서는 홀수 소수 p와 단위원의 p 번째 원시근 $\zeta = \zeta_p$을 고정한다.

SAGE 예 4.4.2. Sage에서는 p 번째 단위원의 해들을 생성하기 위하여 `CyclotomicField`를 사용한다. ζ에 관한 표현은 항상 $p-1$차 이하의 다항식 꼴로 출력한다.

```
sage: K.<zeta> = CyclotomicField(5)
sage: zeta^5
1
sage: 1/zeta
-zeta^3 - zeta^2 - zeta - 1
```

정의 4.4.3 (Gauss 합)**.** 홀수 소수 p를 고정하자. a가 정수일 때

$$g_a = \sum_{n=1}^{p-1} \left(\frac{n}{p}\right) \zeta^{an},$$

를 a와 결합된 **Gauss 합**이라고 한다. 단, $\zeta = \zeta_p = \cos(2\pi/p) + i\sin(2\pi/p) = e^{2\pi i/p}$이다.

g_a의 정의에서 p가 내재되어 있다. 만약 p를 바꾸어야 한다면, a와 결함된 가우스 합은 달라질 것이다. 또 g_a의 정의는 ζ의 선택에도 의존한다. 우리의 정의에서는 $\zeta = \zeta_p$를 선택했지만 다른 ζ를 선택한다면 g_a의 값은 달라질 수 있다.

SAGE 예 4.4.4. 가우스 합을 계산하는 함수 `gauss_sum`를 정의하고 $p = 5$일 때 가우스 합 g_2를 계산한다.

```
sage: def gauss_sum(a,p):
...   K.<zeta> = CyclotomicField(p)
...   return sum(legendre_symbol(n,p) * zeta^(a*n)
...              for n in range(1,p))
sage: g2 = gauss_sum(2,5); g2
2*zeta^3 + 2*zeta^2 + 1
sage: g2.complex_embedding()
-2.23606797749979 + 3.33066907387547e-16*I
sage: g2^2
5
```

여기서 g_2는 ζ_5의 다항식으로 먼저 출력되므로 항상 정확하다. Sage 함수 `complex_embedding`은 g_2의 어떤 복소수로의 매립(embedding)을 보여주는데 처음 15 자릿수까지만 보여준다. $g_2^2 = 5$이므로 $g_2 = -\sqrt{5}$이다.

다음은 가우스 합 g_2를 그림으로 표현하는 Sage코드이다. (Figure 4.1 참조)

```
zeta = CDF(exp(2*pi*I/5))
v = [legendre_symbol(n,5) * zeta^(2*n) for n in range(1,5)]
S = sum([point(tuple(z), pointsize=100) for z in v])
show(S + point(tuple(sum(v)), pointsize=100, rgbcolor='red'))
```

Figure 4.1은 $p = 5$일 때 가우스 합 g_2를 나타내는 그림이다. 가우스 합은 단위원 위의 점들에 정해진 부호를 주어 더하여 얻는다. 이 예는 다음 기초정리를 제안한다. 증명은 몇 가지 작업을 요구한다.

기초정리 4.4.5 (가우스 합)**.** *정수 a가 p의 배수가 아니면 다음 식이 성립한다.*

$$g_a^2 = (-1)^{(p-1)/2} p.$$

SAGE 예 4.4.6. Sage를 이용하여 $p = 7$일 때와 $p = 13$일 때 이 기초정리가 참임을 예로 확인한다.

```
sage: [gauss_sum(a, 7)^2 for a in range(1,7)]
[-7, -7, -7, -7, -7, -7]
sage: [gauss_sum(a, 13)^2 for a in range(1,13)]
[13, 13, 13, 13, 13, 13, 13, 13, 13, 13, 13, 13]
```

이 기초정리를 증명하기 위하여 보조정리가 몇 개 필요하다.

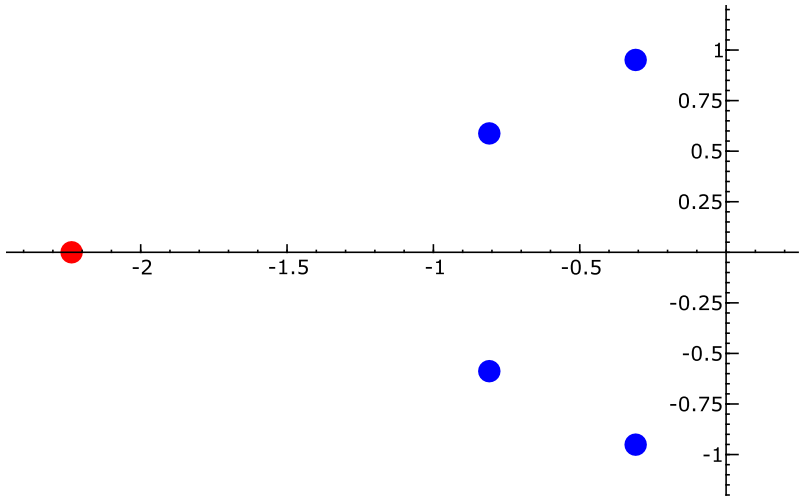


FIGURE 4.1. The red dot is the Gauss sum g_2 for $p = 5$

보조정리 4.4.7. *정수 a에 대하여, 다음이 성립한다.*

$$\sum_{n=0}^{p-1} \zeta^{an} = \begin{cases} p & \text{if } a \equiv 0 \pmod{p}, \\ 0 & \text{otherwise.} \end{cases}$$

증명 만약 $a \equiv 0 \pmod{p}$이면 $\zeta^a = 1$이므로 합은 항의 개수와 같다. 만약 $a \not\equiv 0 \pmod{p}$이면 등식

$$x^p - 1 = (x-1)(x^{p-1} + \cdots + x + 1)$$

를 이용하고 $x = \zeta^a$로 놓자. $\zeta^a \neq 1$이므로 $\zeta^a - 1 \neq 0$이고, 따라서

$$\sum_{n=0}^{p-1} \zeta^{an} = \frac{\zeta^{ap} - 1}{\zeta^a - 1} = \frac{1-1}{\zeta^a - 1} = 0$$

이 성립한다. □

보조정리 4.4.8. *임의의 정수 x, y에 대하여,*

$$\sum_{n=0}^{p-1} \zeta^{(x-y)n} = \begin{cases} p & \text{if } x \equiv y \pmod{p}, \\ 0 & \text{otherwise.} \end{cases}$$

증명 보조정리 4.4.7에서 $a = x - y$로 놓으면 된다. □

보조정리 4.4.9. *$g_0 = 0$이다.*

증명 정의로부터

$$g_0 = \sum_{n=0}^{p-1} \left(\frac{n}{p}\right) \tag{4.4.1}$$

이다. 보조정리 4.1.4에 의해서 함수

$$\left(\frac{\cdot}{p}\right) : (\mathbf{Z}/p\mathbf{Z})^* \to \{\pm 1\}$$

는 군의 전사 준동형사상이다. 그러므로 $(\mathbf{Z}/p\mathbf{Z})^*$의 반은 $+1$로, 나머지 반은 -1에 대응한다. $\left(\frac{0}{p}\right) = 0$이므로, 합 (4.4.1)은 0이다. □

보조정리 4.4.10. *임의의 정수 a에 대하여*

$$g_a = \left(\frac{a}{p}\right) g_1.$$

이다.

증명 $a \equiv 0 \pmod{p}$인 경우는 보조정리 4.4.9로부터 성립하므로 $a \not\equiv 0 \pmod{p}$라고 가정하자. 그러면

$$\left(\frac{a}{p}\right) g_a = \left(\frac{a}{p}\right) \sum_{n=0}^{p-1} \left(\frac{n}{p}\right) \zeta^{an} = \sum_{n=0}^{p-1} \left(\frac{an}{p}\right) \zeta^{an} = \sum_{m=0}^{p-1} \left(\frac{m}{p}\right) \zeta^{m} = g_1$$

가 성립하는데, 세 번째 등식에서 a를 곱하는 것이 $\mathbf{Z}/p\mathbf{Z}$의 일대일 대응 사상임을 이용하였다. 이제 양 변에 $\left(\frac{a}{p}\right)$를 곱하고 $\left(\frac{a}{p}\right)^2 = 1$임을 이용하면 된다. □

이제 보조정리들을 이용하여 기초정리를 4.4.5를 증명할 수 있다.

기초정리 4.4.5의 증명 두 가지 방법으로 $\sum_{a=0}^{p-1} g_a g_{-a}$를 계산한다. 보조정리 4.4.10으로부터, $a \not\equiv 0 \pmod{p}$이므로, 다음이 성립한다.

$$g_a g_{-a} = \left(\frac{a}{p}\right) g_1 \left(\frac{-a}{p}\right) g_1 = \left(\frac{-1}{p}\right) \left(\frac{a}{p}\right)^2 g_1^2 = (-1)^{(p-1)/2} g_1^2$$

이 때 마지막 등호는 기초정리 4.2.1과 $\left(\frac{a}{p}\right) \in \{\pm 1\}$인 사실로부터 얻어진다. 따라서

$$\sum_{a=0}^{p-1} g_a g_{-a} = (p-1)(-1)^{(p-1)/2} g_1^2 \tag{4.4.2}$$

를 얻는다. 한편, 정의로부터

$$\begin{aligned} g_a g_{-a} &= \sum_{n=0}^{p-1} \left(\frac{n}{p}\right) \zeta^{an} \cdot \sum_{m=0}^{p-1} \left(\frac{m}{p}\right) \zeta^{-am} \\ &= \sum_{n=0}^{p-1} \sum_{m=0}^{p-1} \left(\frac{n}{p}\right) \left(\frac{m}{p}\right) \zeta^{an} \zeta^{-am} \\ &= \sum_{n=0}^{p-1} \sum_{m=0}^{p-1} \left(\frac{n}{p}\right) \left(\frac{m}{p}\right) \zeta^{an-am} \end{aligned}$$

이다. $\delta(n,m)$을 $n \equiv m \pmod{p}$이면 1 그렇지 않으면 0이라고 놓자. 그러면 보조정리 4.4.8에 의해서, 다음이 성립한다.

$$\begin{aligned} \sum_{a=0}^{p-1} g_a g_{-a} &= \sum_{a=0}^{p-1} \sum_{n=0}^{p-1} \sum_{m=0}^{p-1} \left(\frac{n}{p}\right) \left(\frac{m}{p}\right) \zeta^{an-am} \\ &= \sum_{n=0}^{p-1} \sum_{m=0}^{p-1} \left(\frac{n}{p}\right) \left(\frac{m}{p}\right) \sum_{a=0}^{p-1} \zeta^{an-am} \\ &= \sum_{n=0}^{p-1} \sum_{m=0}^{p-1} \left(\frac{n}{p}\right) \left(\frac{m}{p}\right) p\delta(n,m) \\ &= \sum_{n=0}^{p-1} \left(\frac{n}{p}\right)^2 p \\ &= p(p-1). \end{aligned}$$

식 (4.4.2)와 위의 식으로부터 $(p-1)$를 소거하면

$$g_1^2 = (-1)^{(p-1)/2} p$$

을 얻는다. $a \not\equiv 0 \pmod{p}$이므로, $\left(\frac{a}{p}\right)^2 = 1$이 되고, 또 보조정리 4.4.10로부터

$$g_a^2 = \left(\frac{a}{p}\right)^2 g_1^2 = g_1^2$$

이므로 이 기초정리의 $g_a^2 = g_1^2 = (-1)^{(p-1)/2} p$이 성립한다. □

4.4.1 이차상호의 증명

이제 가우스 합을 이용하여 정리 4.1.7를 증명한다.

증명 q는 $q \neq p$인 홀수 소수이다. $p^* = (-1)^{(p-1)/2} p$, $g = g_1 = \sum_{n=0}^{p-1} \left(\frac{n}{p}\right) \zeta^n$로 놓으면 기초정리 4.4.5는 $p^* = g^2$임을 기억하자.

기초정리 4.2.1는

$$(p^*)^{(q-1)/2} \equiv \left(\frac{p^*}{q}\right) \pmod{q}$$

를 유도한다. $g^{q-1} = (g^2)^{(q-1)/2} = (p^*)^{(q-1)/2}$이므로 양변에 g를 곱하여 다음 식을 얻는다.

$$g^q \equiv g\left(\frac{p^*}{q}\right) \pmod{q}. \tag{4.4.3}$$

그런데 잠깐! g^q는 정수가 아닌데 이 합동식은 의미하는 것은 무엇인가? 합동식이 의미하는 것은, 양변의 차인 $g^q - g\left(\frac{p^*}{q}\right)$가 환 $\mathbf{Z}[\zeta]$에서 q의 배수라는 것이다. $\mathbf{Z}[\zeta]$는 $\mathbf{Z}$에 계수를 갖는 ζ에 관한 다항식들의 집합이다.

환 $\mathbf{Z}[\zeta]/(q)$는 표수가 q이다. 그래서 만약 $x, y \in \mathbf{Z}[\zeta]$이면, $(x+y)^q \equiv x^q + y^q \pmod{q}$이다. 이것을 (4.4.3)에 적용하면 다음을 얻는다.

$$g^q = \left(\sum_{n=0}^{p-1}\left(\frac{n}{p}\right)\zeta^n\right)^q \equiv \sum_{n=0}^{p-1}\left(\frac{n}{p}\right)^q \zeta^{nq} \equiv \sum_{n=0}^{p-1}\left(\frac{n}{p}\right)\zeta^{nq} \equiv g_q \pmod{q}.$$

보조정리 4.4.10에 의해서,

$$g^q \equiv g_q \equiv \left(\frac{q}{p}\right) g \pmod{q}$$

를 얻고, 이를 g^q에 적용하면 다음 식을 얻는다.

$$\left(\frac{q}{p}\right) g \equiv \left(\frac{p^*}{q}\right) g \pmod{q}.$$

$g^2 = p^*$ and $p \neq q$이므로 양 변의 g를 소거하면 $\left(\frac{q}{p}\right) \equiv \left(\frac{p^*}{q}\right) \pmod{q}$를 얻는다. 두 잉여부호는 ± 1이고 q는 홀수이므로 $(1 \equiv -1 \pmod{q} \iff q = 2)$, $\left(\frac{q}{p}\right)$와 $\left(\frac{p^*}{q}\right)$는 둘 다 1이거나 -1이어야만 되므로 $\left(\frac{q}{p}\right) = \left(\frac{p^*}{q}\right)$이다. 마지막으로 따름정리 4.2.3를 사용하면

$$\left(\frac{p^*}{q}\right) = \left(\frac{(-1)^{(p-1)/2}p}{q}\right) = \left(\frac{-1}{q}\right)^{(p-1)/2}\left(\frac{p}{q}\right) = (-1)^{\frac{q-1}{2}\cdot\frac{p-1}{2}} \cdot \left(\frac{p}{q}\right)$$

를 얻으므로 이차잉여가 증명된다. □

4.5 제곱근 찾기

이 절에서는 제곱근을 찾는 문제를 다시 다룬다. $2 \neq 0$인 체 K에서는 $a \neq 0$인 모든 $a, b, c \in K$에 대하여 이차방정식 $ax^2 + bx + c = 0$의 두 해는 다음과 같다.

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}.$$

이제 p가 홀수 소수이고 $K = \mathbf{Z}/p\mathbf{Z}$라 하자. 정리 4.1.7를 사용하면 $b^2 - 4ac$가 $\mathbf{Z}/p\mathbf{Z}$에서 완전제곱인지 아닌지를 알 수 있고, 그러므로 $ax^2 + bx + c = 0$가 $\mathbf{Z}/p\mathbf{Z}$에서 해를 갖는지 아닌지를 알수 있다. 그러나 정리 4.1.7은 해가 존재한다 하더라도 실제 해를 찾는 방법에 대해서는 어떤 정보도 주지 않는다. 물론 $\mathbf{Z}/p\mathbf{Z}$의 원소가 완전 제곱인지 아니지를 확인하는 실제 계산에서는 이차잉여법칙이 다 필요하지도 않다. 오히려 절 2.3의 관점에서 기초정리 4.2.1를 이용하는 것이 상당히 빠르다.

$a \in \mathbf{Z}/p\mathbf{Z}$가 0이 아닌 이차 잉여라고 가정하자. 만약 $p \equiv 3 \pmod 4$이면, $b = a^{\frac{p+1}{4}}$는

$$b^2 = a^{\frac{p+1}{2}} = a^{\frac{p-1}{2}+1} = a^{\frac{p-1}{2}} \cdot a = \left(\frac{a}{p}\right) \cdot a = a$$

을 만족하므로 $b = a^{\frac{p+1}{4}}$는 a의 제곱근이다. 따라서 절 2.3의 거듭제곱 알고리듬을 이용하여 p의 자릿수의 다항식 시간 안에 b를 계산할 수 있다.

다음은 $p \equiv 1 \pmod 4$라고 가정하자. 불행히도 이 경우에는 a와 p를 입력했을 때 법 p에서 a의 제곱근이 존재하더라도 이 제곱근을 출력하는 결정적인 $\log(p)$의 다항식 시간 알고리듬을 알지 못한다.

참조 4.5.1. Schoof [45]덕분에 $O((\sqrt{(}|a|)^{1/2+\varepsilon} \cdot \log(p))^9)$ 시간 안에 a의 제곱근을 계산하는 알고리듬은 존재한다. 이 멋진 알고리듬은 (타원곡선을 사용하는데) 위에서 언급한 의미에서의 다항식 시간의 알고리듬은 아니다. 왜냐하면 a가 커지면 a가 작은 수일 때보다는 지수적으로 더 긴 시간이 필요하다.

다음은 법 p에서 a의 제곱근을 상당히 빨리 계산하는 확률적인 알고리듬을 소개한다. 정의 2.1.3로부터 환의 개념을 기억하기 바란다. 또 환의 준동형사상과 동형사상의 개념도 필요하다.

정의 4.5.2 (환의 준동형사상)**.** R과 S가 환일 때 함수 $\varphi : R \to S$ 가 모든 $a, b \in R$에 대해서

- $\varphi(ab) = \varphi(a)\varphi(b)$,
- $\varphi(a + b) = \varphi(a) + \varphi(b)$, and
- $\varphi(1) = 1$

를 만족하면 φ를 **환의 준동형사상**(homomorphism of rings)이라 부른다. **동형사상(isomorphism)**은 일대일 대응인 준동형사상 $\varphi : R \to S$을 의미한다.

환

$$R = (\mathbf{Z}/p\mathbf{Z})[x]/(x^2 - a)$$

을 다음과 같이 정의한다. 집합으로서는

$$R = \{u + v\alpha : u, v \in \mathbf{Z}/p\mathbf{Z}\}$$

이고 곱셈은

$$(u + v\alpha)(z + w\alpha) = (uz + awv) + (uw + vz)\alpha$$

로 정의한다. 여기서 α는 R에서 x의 잉여류이다.

SAGE 예 4.5.3. 위에서 정의한 환 R은 Sage에서는 다음과 같이 정의하고 연산을 수행한다. ($p = 13$인 경우):

```
sage: S.<x> = PolynomialRing(GF(13))
sage: R.<alpha> = S.quotient(x^2 - 3)
sage: (2+3*alpha)*(1+2*alpha)
7*alpha + 7
```

b와 c가 $\mathbf{Z}/p\mathbf{Z}$에서 a의 제곱근이라 하자. (비록 b와 c를 아직은 쉽게 계산할 수는 없지만, 제곱근을 찾는 알고리듬을 유도하기 위하여 제곱근 b와 c를 고려한다.) 그러면 각각 $f(u+v\alpha) = u + vb$와 $g(u+v\alpha) = u + vc$로 정의된 두 개의 환 준동형사상 $f : R \to \mathbf{Z}/p\mathbf{Z}$와 $g : R \to \mathbf{Z}/p\mathbf{Z}$ 이 존재한다. 이 두 함수는 함께 환 준동형사상

$$\varphi : R \longrightarrow \mathbf{Z}/p\mathbf{Z} \times \mathbf{Z}/p\mathbf{Z}$$

를 유도한다. 이 때 $\varphi(u+v\alpha) = (u+vb, u+vc)$이다. 적당한 방법을 사용하여 $(\mathbf{Z}/p\mathbf{Z})^*$의 임의의 원소 z를 선택하여 절 2.3.2의 이진법 거듭제곱 알고리듬으로 $(1+z\alpha)^{\frac{p-1}{2}}$를 재빨리 계산한 후

$$u + v\alpha = (1+z\alpha)^{\frac{p-1}{2}}$$

를 만족하는 $u, v \in \mathbf{Z}/p\mathbf{Z}$를 찾는다. 만약 $v = 0$이면 다른 z를 무작위로 선택하여 같은 계산을 반복한다. 만약 $v \neq 0$이면, 우리가 구하고자 하는 제곱근 b와 c를 다음과 같이 빠르게 찾을 수 있다. $u + vb$는 $\mathbf{Z}/p\mathbf{Z}$에서 어떤 원소의 $(p-1)/2$ 거듭제곱이므로 이 값은 0, 1, 혹은 -1이다. 따라서 각 경우 $b = -u/v$, $(1-u)/v$, 혹은 $(-1-u)/v$이다. 우리는 u와 v를 알기 때문에, $-u/v$, $(1-u)/v$, 그리고 $(-1-u)/v$의 제곱들을 계산하여 a의 제곱근인지를 확인하면 된다.

예 4.5.4. 예 4.1.8의 계속으로 법 389에서 69의 제곱근을 구한다. 위에서 설명한 알고리듬을 $p \equiv 1 \pmod 4$인 경우에 적용한다. 먼저 무작위로 $z = 24$로 선택하여 $(1+24\alpha)^{194}$를 계산하면 $(1+24\alpha)^{194} = -1$를 얻는다. α의 계수가 0이므로 $z = 51$로 $(1+51\alpha)^{194}$를 계산한다. 이번에는 $(1+51\alpha)^{194} = 239\alpha = u + v\alpha$로 α의 계수가 0이 아니다. $\mathbf{Z}/389\mathbf{Z}$에서 239 의 역원은 153이므로, 69의 제곱근이 될 수 있는 세 개의 원소는 다음과 같다.

$$-\frac{u}{v} = 0 \qquad \frac{1-u}{v} = 153 \qquad -\frac{1-u}{v} = -153.$$

따라서 153과 -153이 $\mathbf{Z}/389\mathbf{Z}$에서 69의 제곱근이다.

SAGE 예 4.5.5. 위의 제곱근을 구하는 알고리듬을 Sage에서 구현하고 몇 가지 예들을 보여준다.

```
sage: def find_sqrt(a, p):
...     assert (p-1)%4 == 0
...     assert legendre_symbol(a,p) == 1
...     S.<x> = PolynomialRing(GF(p))
```

```
...     R.<alpha> = S.quotient(x^2 - a)
...     while True:
...         z = GF(p).random_element()
...         w = (1 + z*alpha)^((p-1)//2)
...         (u, v) = (w[0], w[1])
...         if v != 0: break
...     if (-u/v)^2 == a: return -u/v
...     if ((1-u)/v)^2 == a: return (1-u)/v
...     if ((-1-u)/v)^2 == a: return (-1-u)/v
...
sage: b = find_sqrt(3,13)
sage: b                         # random: either 9 or 3
9
sage: b^2
3
sage: b = find_sqrt(3,13)
sage: b                         # see, it's random
4
sage: find_sqrt(5,389)          # random: either 303 or 86
303
sage: find_sqrt(5,389)          # see, it's random
86
```

4.6 Exercises

4.1 Calculate the following by hand: $\left(\frac{3}{97}\right)$, $\left(\frac{3}{389}\right)$, $\left(\frac{22}{11}\right)$, and $\left(\frac{5!}{7}\right)$.

4.2 Let G be an abelian group, and let n be a positive integer.

(a) Prove that the map $\varphi : G \to G$ given by $\varphi(x) = x^n$ is a group homomorphism.

(b) Prove that the subset H of G of squares of elements of G is a subgroup.

4.3 Use Theorem 4.1.7 to show that for $p \geq 5$ prime,

$$\left(\frac{3}{p}\right) = \begin{cases} 1 & \text{if } p \equiv 1, 11 \pmod{12}, \\ -1 & \text{if } p \equiv 5, 7 \pmod{12}. \end{cases}$$

4.4 (*) Use that $(\mathbf{Z}/p\mathbf{Z})^*$ is cyclic to give a direct proof that $\left(\frac{-3}{p}\right) = 1$ when $p \equiv 1 \pmod 3$. (Hint: There is an element $c \in (\mathbf{Z}/p\mathbf{Z})^*$ of order 3. Show that $(2c+1)^2 = -3$.)

4.5 (*) If $p \equiv 1 \pmod 5$, show directly that $\left(\frac{5}{p}\right) = 1$ by the method of Exercise 4.4. (Hint: Let $c \in (\mathbf{Z}/p\mathbf{Z})^*$ be an element of order 5. Show that $(c + c^4)^2 + (c + c^4) - 1 = 0$, etc.)

4.6 (*) Let p be an odd prime. In this exercise, you will prove that $\left(\frac{2}{p}\right) = 1$ if and only if $p \equiv \pm 1 \pmod 8$.

(a) Prove that

$$x = \frac{1 - t^2}{1 + t^2}, \qquad y = \frac{2t}{1 + t^2}$$

is a parameterization of the set of solutions to $x^2 + y^2 \equiv 1 \pmod p$, in the sense that the solutions $(x, y) \in \mathbf{Z}/p\mathbf{Z}$ are in bijection with the $t \in \mathbf{Z}/p\mathbf{Z} \cup \{\infty\}$ such that $1 + t^2 \not\equiv 0 \pmod p$. Here, $t = \infty$ corresponds to the point $(-1, 0)$. (Hint: if (x_1, y_1) is a solution, consider the line $y = t(x + 1)$ through (x_1, y_1) and $(-1, 0)$, and solve for x_1, y_1 in terms of t.)

(b) Prove that the number of solutions to $x^2 + y^2 \equiv 1 \pmod p$ is $p + 1$ if $p \equiv 3 \pmod 4$ and $p - 1$ if $p \equiv 1 \pmod 4$.

(c) Consider the set S of pairs $(a, b) \in (\mathbf{Z}/p\mathbf{Z})^* \times (\mathbf{Z}/p\mathbf{Z})^*$ such that $a + b = 1$ and $\left(\frac{a}{p}\right) = \left(\frac{b}{p}\right) = 1$. Prove that $\#S = (p + 1 - 4)/4$ if $p \equiv 3 \pmod 4$ and $\#S = (p - 1 - 4)/4$ if $p \equiv 1 \pmod 4$. Conclude that $\#S$ is odd if and only if $p \equiv \pm 1 \pmod 8$.

(d) The map $\sigma(a, b) = (b, a)$ that swaps coordinates is a bijection of the set S. It has exactly one fixed point if and only if there is an $a \in \mathbf{Z}/p\mathbf{Z}$ such that $2a = 1$ and $\left(\frac{a}{p}\right) = 1$. Also, prove that $2a = 1$ has a solution $a \in \mathbf{Z}/p\mathbf{Z}$ with $\left(\frac{a}{p}\right) = 1$ if and only if $\left(\frac{2}{p}\right) = 1$.

(e) Finish by showing that σ has exactly one fixed point if and only if $\#S$ is odd, i.e., if and only if $p \equiv \pm 1 \pmod 8$.

Remark: The method of proof of this exercise can be generalized to give a proof of the full Quadratic Reciprocity Law.

4.7 How many natural numbers $x < 2^{13}$ satisfy the equation

$$x^2 \equiv 5 \pmod{2^{13} - 1}?$$

You may assume that $2^{13} - 1$ is prime.

4.8 Find the natural number $x < 97$ such that $x \equiv 4^{48} \pmod{97}$. Note that 97 is prime.

4.9 In this problem, we will formulate an analog of quadratic reciprocity for a symbol like $\left(\frac{a}{q}\right)$, but without the restriction that q be a prime. Suppose n is an odd positive integer, which we factor as $\prod_{i=1}^{k} p_i^{e_i}$. We define the Jacobi symbol $\left(\frac{a}{n}\right)$ as follows:

$$\left(\frac{a}{n}\right) = \prod_{i=1}^{k} \left(\frac{a}{p_i}\right)^{e_i}.$$

(a) Give an example to show that $\left(\frac{a}{n}\right) = 1$ need not imply that a is a perfect square modulo n.

(b) (*) Let n be odd and a and b be integers. Prove that the following holds:

i. $\left(\frac{a}{n}\right)\left(\frac{b}{n}\right) = \left(\frac{ab}{n}\right)$. (Thus $a \mapsto \left(\frac{a}{n}\right)$ induces a homomorphism from $(\mathbf{Z}/n\mathbf{Z})^*$ to $\{\pm 1\}$.)

ii. $\left(\frac{-1}{n}\right) \equiv n \pmod 4$.

iii. $\left(\frac{2}{n}\right) = 1$ if $n \equiv \pm 1 \pmod 8$ and -1 otherwise.

iv. Assume a is positive and odd. Then $\left(\frac{a}{n}\right) = (-1)^{\frac{a-1}{2}\cdot\frac{n-1}{2}}\left(\frac{n}{a}\right)$

4.10 (*) Prove that for any $n \in \mathbf{Z}$, the integer $n^2 + n + 1$ does not have any divisors of the form $6k - 1$.

5
연분수
(Continued Fractions)

황금비 $\frac{1+\sqrt{5}}{2}$는 무한 분수형태인

$$1+\cfrac{1}{1+\cfrac{1}{1+\cfrac{1}{1+\cdots}}}$$

와 같고, 다음 분수

$$\frac{103993}{33102} = 3.14159265301190260407\ldots$$

는 π의 아주 멋진 분수 근사값이다. 이 두 관찰 모두 연분수로 설명할 수 있다.

연분수는 이론적으로 아름다울뿐만 아니라 정수론의 문제들을 해결하는데 필요한 강력한 알고리듬을 만들어내는데 필요한 도구를 제공한다. 예를 들어 연분수는 심지어 몇 백 자릿수가 되는 소수가, 두 제곱수의 합이 되는 경우, 두 제곱수를 합으로 표현하는 빠른 방법을 제공한다.

따라서 연분수는 계산면에서나 개념적인 면에서 응용이 많은 정수론의 멋진 도구이다. 예를 들어 연분수는 소수로 표현된 분수의 처음 몇 개의 소숫점 자릿수들로부터 원래의 분수를 인식하는 놀랄만큼 효율적인 방법을 제공한다. 또 연분수는 e가 π보다 덜 복잡하다고 감각적으로 느낄 수 있도록 한다. (예 5.3.4와 5.4절 참조)

5.2절에서 우리는 먼저 유한연분수를 공부하고 앞으로의 조사를 위한 기초를 마련한다. 5.3절에서는 실수 x에 x로 수렴하는 연분수를 찾는 과정을 알아본다. 5.5절에서는, 순환연분수는 바로 이차방정식의 무리수 해로서 특징지어지는 것을 확인하고, 차수가 2보다 큰 기약다항식의 해의 연분수에 관한

미해결 문제에 대해 토론한다. 마지막으로 연분수를 이용하여 유리수의 근사값을 인식하고(5.6절), 정수를 두 제곱수의 합으로 쓰는 것에 응용하는 것을 소개하면서(5.7절) 이 단원을 마무리한다. 독자들에게는 참고문헌 [23, Ch. X], [26], [5, §13.3], [39, Ch. 7]에서 연분수 부분을 더 읽어보기를 권장한다.

5.1 정의

연분수(continued fraction)는

$$a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{a_3 + \cdots}}}.$$

와 같이 표현되는 분수이다. 이 책에서는 a_i는 실수이고 $i \geq 1$일 때 $a_i > 0$으로 가정하고, 또 위의 분수 표현은 유한할 수도 무한할 수도 있다. 연분수의 더 일반적인 개념들도 광범위하게 연구되었지만 대부분은 우리 책의 범위를 벗어난다. 이 책에서는 a_i가 모두 정수인 경우에 가장 관심을 가지려고 한다.

우리는 위의 연분수를 간단히

$$[a_0, a_1, a_2, \ldots]$$

로 표현하자. 예를 들어

$$[1,2] = 1 + \frac{1}{2} = \frac{3}{2},$$

$$\begin{aligned}[3,7,15,1,292] &= 3 + \cfrac{1}{7 + \cfrac{1}{15 + \cfrac{1}{1 + \cfrac{1}{292}}}} \\ &= \frac{103993}{33102} = 3.14159265301190260407\ldots,\end{aligned}$$

$$\begin{aligned}[2,1,2,1,1,4,1,1,6] &= 2 + \cfrac{1}{1 + \cfrac{1}{2 + \cfrac{1}{1 + \cfrac{1}{1 + \cfrac{1}{4 + \cfrac{1}{1 + \cfrac{1}{1 + \cfrac{1}{6}}}}}}}} \\ &= \frac{1264}{465} \\ &= 2.71827956989247311827956989\ldots\end{aligned}$$

등이다. 두 번째 세 번째 예는 연분수는 무리수의 유리수 근사값을 구하는데 이용될 수 있다는 것을 미리 짐작할 수 있도록 선택하였다. 앞의 것은 π로, 뒤의 것은 e로 접근하는 유리수이다.

5.2 유한연분수

이 절은 $m \geq 0$일 때 $[a_0, a_1, \ldots, a_m]$꼴의 연분수에 관한 절이다. 먼저 모든 $n \leq m$에 대하여

$$[a_0, a_1, \ldots, a_n] = \frac{p_n}{q_n} \tag{5.2.1}$$

이 성립하는 p_n과 q_n을 귀납적으로 정의한다.

그런 후 부분수렴 $[a_0, \ldots, a_k]$들의 수열에 대한 여러 성질들을 추론할 때 반복적으로 사용하는 2×2 행렬 $\left(\begin{smallmatrix} p_n & p_{n-1} \\ q_n & q_{n-1} \end{smallmatrix}\right)$와 $\left(\begin{smallmatrix} p_n & p_{n-2} \\ q_n & q_{n-2} \end{smallmatrix}\right)$의 행렬식에 관련된 공식을 찾아낸다.

모든 유리수는, 식 5.2.1에서처럼, 연분수로 표현된다는 것을 증명하기 위하여 알고리듬 1.1.13을 사용한다.

정의 5.2.1 (유한연분수)**. 유한연분수(finite continued fraction)**는 다음과 같은 분수이다.

$$a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{\cdots + \frac{1}{a_n}}}}$$

이 때 각 a_m은 실수이고, 모든 $m \geq 1$에 대해서 $a_m > 0$이다.

정의 5.2.2 (단순연분수)**. 단순연분수(simple continued fraction)**는 모든 a_i가 정수인 유한, 혹은 무한 연분수이다.

연분수의 값에 대한 감을 얻기 위하여, 다음 값들을 관찰하자.

$$\begin{aligned} [a_0] &= a_0, \\ [a_0, a_1] &= a_0 + \frac{1}{a_1} = \frac{a_0 a_1 + 1}{a_1}, \\ [a_0, a_1, a_2] &= a_0 + \frac{1}{a_1 + \frac{1}{a_2}} = \frac{a_0 a_1 a_2 + a_0 + a_2}{a_1 a_2 + 1}. \end{aligned}$$

또,

$$\begin{aligned} [a_0, a_1, \ldots, a_{n-1}, a_n] &= \left[a_0, a_1, \ldots, a_{n-2}, a_{n-1} + \frac{1}{a_n}\right] \\ &= a_0 + \frac{1}{[a_1, \ldots, a_n]} \\ &= [a_0, [a_1, \ldots, a_n]]. \end{aligned}$$

SAGE 예 5.2.3. `continued_fraction`은 연분수를 계산하는 sage 함수이다.

```
sage: continued_fraction(17/23)
[0, 1, 2, 1, 5]
sage: continued_fraction(e)
[2, 1, 2, 1, 1, 4, 1, 1, 6, 1, 1, 8, 1, 1, 10, 1, 1,
 12, 1, 1]
```

입력한 수의 연분수를 계산할 때 정확성(in bits)을 결정하기 위하여 옵션으로 `bits = n`을 사용한다.

```
sage: continued_fraction(e, bits=21)
[2, 1, 2, 1, 1, 4, 1, 1, 6]
sage: continued_fraction(e, bits=30)
[2, 1, 2, 1, 1, 4, 1, 1, 6, 1, 1, 8]
```

또 연분수의 값도 구할 수 있고, 심지어 연분수를 가지고 계산도 할 수 있다.

```
sage: a = continued_fraction(17/23); a
[0, 1, 2, 1, 5]
sage: a.value()
17/23
sage: b = continued_fraction(6/23); b
[0, 3, 1, 5]
sage: a + b
[1]
```

5.2.1 부분수렴

유한 연분수 $[a_0, \ldots, a_m]$를 하나 고정하자. 지금은 a_i를 정수라고 가정할 필요는 없다.

정의 5.2.4 (부분수렴)**.** $0 \le n \le m$을 만족하는 n에 대하여, 연분수 $[a_0, \ldots, a_n]$을 연분수 $[a_0, \ldots, a_m]$의 n번 째 **부분수렴(partial convergent)**이라 부른다.

$-2 \le n \le m$을 만족하는 각각의 n에 대하여, 실수 p_n과 q_n을 다음과 같이 정의:

$$\begin{array}{llll} p_{-2} = 0, & p_{-1} = 1, & p_0 = a_0, & \cdots \quad p_n = a_n p_{n-1} + p_{n-2} \quad \cdots, \\ q_{-2} = 1, & q_{-1} = 0, & q_0 = 1, & \cdots \quad q_n = a_n q_{n-1} + q_{n-2} \quad \cdots. \end{array}$$

기초정리 5.2.5 (부분수렴))**.** *$n \le m$인 $n \ge 0$에 대하여,*

$$[a_0, \ldots, a_n] = \frac{p_n}{q_n}$$

이 성립한다.

증명 귀납법을 사용하자. $n = 0, 1$일 때는 자명하다. 위의 기초정리가 길이가 $n-1$인 연분수에서는 항상 성립한다고 가정하자. 그러면 다음이 성립한다.

$$\begin{aligned}
[a_0, \ldots, a_n] &= [a_0, \ldots, a_{n-2}, a_{n-1} + \frac{1}{a_n}] \\
&= \frac{\left(a_{n-1} + \frac{1}{a_n}\right) p_{n-2} + p_{n-3}}{\left(a_{n-1} + \frac{1}{a_n}\right) q_{n-2} + q_{n-3}} \\
&= \frac{(a_{n-1}a_n + 1)p_{n-2} + a_n p_{n-3}}{(a_{n-1}a_n + 1)q_{n-2} + a_n q_{n-3}} \\
&= \frac{a_n(a_{n-1}p_{n-2} + p_{n-3}) + p_{n-2}}{a_n(a_{n-1}q_{n-2} + q_{n-3}) + q_{n-2}} \\
&= \frac{a_n p_{n-1} + p_{n-2}}{a_n q_{n-1} + q_{n-2}} \\
&= \frac{p_n}{q_n}.
\end{aligned}$$

□

SAGE 예 5.2.6. 만약 c가 연분수를 나타낸다면, c의 모든 부분수렴을 얻으려면 c.convergents()를 사용한다.

```
sage: c = continued_fraction(pi,bits=35); c
[3, 7, 15, 1, 292, 1]
sage: c.convergents()
[3, 22/7, 333/106, 355/113, 103993/33102, 208341/66317]%104348/33215--tex file 오류
```

곧 정확한 개념으로 확인하겠지만, 연분수의 부분수렴값들은 연분수의 최적의 유리수 근사값들이다. 위의 예에서는 연분수 $[3, 7, 15, 1, 292, 1]$의 부분수렴값들은, 주어진 분모의 크기에서는 이 연분수의 가장 정확한 분수 근사값이다.

기초정리 5.2.7. *$n \leq m$인 모든 $n \geq 0$에서 다음 두 식이 성립한다.*

$$p_n q_{n-1} - q_n p_{n-1} = (-1)^{n-1}, \tag{5.2.2}$$

$$p_n q_{n-2} - q_n p_{n-2} = (-1)^n a_n. \tag{5.2.3}$$

또는, 위의 두 식들과 동치인, 다음 두 식이 성립한다.

$$\frac{p_n}{q_n} - \frac{p_{n-1}}{q_{n-1}} = (-1)^{n-1} \cdot \frac{1}{q_n q_{n-1}},$$

$$\frac{p_n}{q_n} - \frac{p_{n-2}}{q_{n-2}} = (-1)^n \cdot \frac{a_n}{q_n q_{n-2}}.$$

증명 $n=0$인 경우는 정의로부터 자명하다. 이제 $n>0$이라고 가정하고 $n-1$일 때 성립한다고 가정하자. 그러면

$$\begin{aligned} p_nq_{n-1}-q_np_{n-1} &= (a_np_{n-1}+p_{n-2})q_{n-1}-(a_nq_{n-1}+q_{n-2})p_{n-1} \\ &= p_{n-2}q_{n-1}-q_{n-2}p_{n-1} \\ &= -(p_{n-1}q_{n-2}-p_{n-2}q_{n-1}) \\ &= -(-1)^{n-2} = (-1)^{n-1}. \end{aligned}$$

이 성립하고, (5.2.2)이 증명된다. 식 (5.2.3)의 증명하기 위해서는 다음을 계산한다.

$$\begin{aligned} p_nq_{n-2}-p_{n-2}q_n &= (a_np_{n-1}+p_{n-2})q_{n-2}-p_{n-2}(a_nq_{n-1}+q_{n-2}) \\ &= a_n(p_{n-1}q_{n-2}-p_{n-2}q_{n-1}) \\ &= (-1)^n a_n \end{aligned}$$

□

참조 5.2.8. 위의 기초정리의 식들을 행렬로 표현한다면 행렬 $\left(\begin{smallmatrix} p_n & p_{n-1} \\ q_n & q_{n-1}\end{smallmatrix}\right)$의 행렬식은 $(-1)^{n-1}$, $\left(\begin{smallmatrix} p_n & p_{n-2} \\ q_n & q_{n-2}\end{smallmatrix}\right)$의 행렬식은 $(-1)^n a_n$이다.

SAGE 예 5.2.9. π의 연분수의 처음 몇 개의 항으로, 기초정리 5.2.7가 참임을 Sage를 이용하여 확인하자.

```
sage: c = continued_fraction(pi); c
[3, 7, 15, 1, 292, 1, 1, 1, 2, 1, 3, 1, 14]
sage: for n in range(-1, len(c)):
...     print c.pn(n)*c.qn(n-1) - c.qn(n)*c.pn(n-1),
1 -1 1 -1 1 -1 1 -1 1 -1 1 -1 1 -1
sage: for n in range(len(c)):
...     print c.pn(n)*c.qn(n-2) - c.qn(n)*c.pn(n-2),
3 -7 15 -1 292 -1 1 -1 2 -1 3 -1 14
```

따름정리 5.2.10 (기약분수로서의 부분수렴)**.** *만약* $[a_0,a_1,\ldots,a_m]$*이 단순연분수라면,* p_n*와* q_n*는 서로 소인 정수이다. 즉,* p_n/q_n*은 기약분수이다.*

증명 p_n과 q_n의 정의로부터 p_n과 q_n이 정수임은 자명하다. d가 p_n과 q_n의 공약수라면, $d \mid p_nq_{n-1}-q_np_{n-1}=(-1)^{n-1}$이므로 $d=1$이다. □

SAGE 예 5.2.11. 따름정리 5.2.10를 Sage를 이용하여 확인한다.

```
sage: c = continued_fraction([1,2,3,4,5])
sage: c.convergents()
[1, 3/2, 10/7, 43/30, 225/157]
sage: [c.pn(n) for n in range(len(c))]
[1, 3, 10, 43, 225]
sage: [c.qn(n) for n in range(len(c))]
[1, 2, 7, 30, 157]
```

5.2.2 부분수렴들의 수열

$[a_0, \ldots, a_m]$은 연분수이고, $n \leq m$인 n에 대하여,

$$c_n = [a_0, \ldots, a_n] = \frac{p_n}{q_n}$$

은 n번 째 부분수렴이라 하자. 연분수의 정의로부터 $n > 0$일 때는 $a_n > 0$이었음을 기억하자. 이 조건이 연분수의 부분수렴값들에 멋진 조건을 더해준다. 예를 들어 $[2, 1, 2, 1, 1, 4, 1, 1, 6]$의 부분수렴들을 나열하면

$$2, 3, 8/3, 11/4, 19/7, 87/32, 106/39, 193/71, 1264/465$$

이다. 이 수들의 크기가 좀 더 잘 보이게 하기 위하여 소수로 표현해보자. 또 다른 구조를 잘 볼 수 있도록, 한 숫자씩 건너 뛰면서 밑줄을 쳐 두었다.

$$\underline{2}, 3, \underline{2.66667}, 2.75000, \underline{2.71429}, 2.71875, \underline{2.71795}, 2.71831, \underline{2.71828}$$

밑줄친 수들은 밑줄을 치지 않은 모든 수들보다 작고, 밑줄 친 수들은 단조증가수열을 이루고, 반면에 밑줄을 치지 않은 수들을 단조감소수열을 이룬다.

SAGE 예 5.2.12. 그림 5.1은 Sage를 이용하여 연분수의 위의 패턴을 또 다른 연분수로 확인한 그림이다.

```
sage: c = continued_fraction([1,1,1,1,1,1,1,1])
sage: v = [(i, c.pn(i)/c.qn(i)) for i in range(len(c))]
sage: P = point(v, rgbcolor=(0,0,1), pointsize=40)
sage: L = line(v, rgbcolor=(0.5,0.5,0.5))
sage: L2 = line([(0,c.value()),(len(c)-1,c.value())], \
...       thickness=0.5, rgbcolor=(0.7,0,0))
sage: (L+L2+P).show(xmin=0,ymin=1)
```

다음은 지금 예들로 확인한 이 현상을 일반적으로 증명한다.

기초정리 5.2.13 (부분수렴은 어떻게 수렴하나)**.** *짝수번째 부분수렴 c_{2n}은 단순증가수열을, 홀수번째 부분수렴 c_{2n+1}은 단조감소수열을 이룬다. 또 임의의 홀수번째 부분수렴 c_{2n+1}은 임의의 짝수번째 부분수렴 c_{2m}보다 크다.*

증명 $n \geq 1$이면 a_n은 양수이므로, q_n은 양수이다. 기초정리 5.2.7에 의하여, $n \geq 2$이면,

$$c_n - c_{n-2} = (-1)^n \cdot \frac{a_n}{q_n q_{n-2}}$$

이 성립하므로, 처음 주장은 성립한다.

두 번째 주장이 성립하지 않는다면, $c_{2m+1} < c_{2r}$을 만족하는 정수 r과 m이 존재한다. 그러면 기초정리 5.2.7에 의하여, $n \geq 1$이면,

$$c_n - c_{n-1} = (-1)^{n-1} \cdot \frac{1}{q_n q_{n-1}}$$

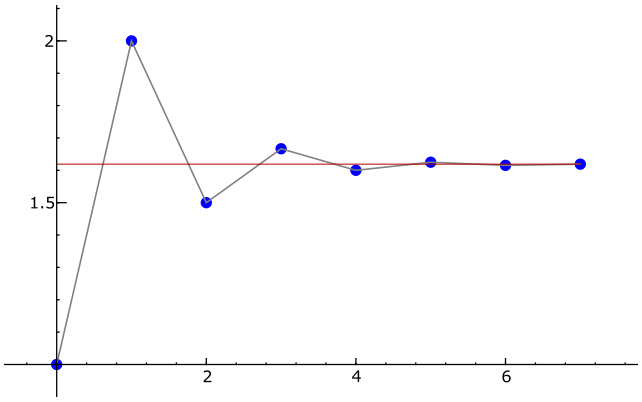


FIGURE 5.1. Graph of a Continued Fraction

은 부호가 $(-1)^{n-1}$이므로, 모든 $s \geq 0$에 대하여 $c_{2s+1} > c_{2s}$이다. 따라서 $r = m$은 불가능하다. 만약 $r < m$이라면, 이 기초정리의 첫 번째 주장으로부터 $c_{2m+1} < c_{2r} < c_{2m}$이 되어 역시 불가능하다. $r > m$이면, $c_{2r+1} < c_{2m+1} < c_{2r}$가 되는데 역시 모순이다. 따라서 $c_{2m+1} < c_{2r}$을 만족하는 정수 r과 m은 존재하지 않는다. □

5.2.3 모든 유리수의 연분수 표현

기초정리 5.2.14 (유리수의 연분수)**.** *모든 영이 아닌 유리수는 단순연분수로 표현된다.*

증명 일반성을 잃지 않고, 유리수 a/b에서 $b \geq 1$ 이고 $\gcd(a,b) = 1$라고 가정하자. 알고리듬 1.1.13로부터

$$\begin{aligned}
a &= b \cdot a_0 + r_1, & 0 < r_1 < b \\
b &= r_1 \cdot a_1 + r_2, & 0 < r_2 < r_1 \\
&\cdots \\
r_{n-2} &= r_{n-1} \cdot a_{n-1} + r_n, & 0 < r_n < r_{n-1} \\
r_{n-1} &= r_n \cdot a_n + 0.
\end{aligned}$$

로 쓸 수 있다. $i > 0$일 $a_i > 0$이고, 또 $\gcd(a, b) = 1$이므로 $r_n = 1$임을 기억하다. 위의 식들은 분수형태로 쓰면 다음과 같이 쓸 수 있다.

$$\begin{aligned} a/b &= a_0 + r_1/b = a_0 + 1/(b/r_1), \\ b/r_1 &= a_1 + r_2/r_1 = a_1 + 1/(r_1/r_2), \\ r_1/r_2 &= a_2 + r_3/r_2 = a_2 + 1/(r_2/r_3), \\ &\cdots \\ r_{n-1}/r_n &= a_n. \end{aligned}$$

위의 식들로부터

$$\frac{a}{b} = [a_0, a_1, \ldots, a_n]$$

임을 알 수 있다. □

기초정리 5.2.14의 증명으로부터 유리수의 연분수를 계산하는 알고리듬을 얻는다.

영이 아닌 유리수는 정확히 두 가지 형태의 연분수로 표현할 수 있다. 예를 들어, $2 = [1, 1] = [2]$ (Exercise 5.2 참조).

5.3 무한연분수

이 절은 실수 x에 정수들의 수열 $a_0, a_1, \ldots$을 연결하는 연분수 계산 과정으로 시작한다. 여러 예를 먼저 관찰한 후 연분수의 홀수 번째와 짝수 번째 부분수렴이 한없이 가까와짐을 보임으로써 $x = \lim_{n\to\infty}[a_0, a_1, \ldots, a_n]$임을 증명한다. 또한 $a_0, a_1, \ldots$이 양의 정수들의 무한 수열이면 $c_n = [a_0, a_1, \ldots, a_n]$은 수렴함을 확인한다. 사실, a_n이 임의의 양수들의 수열인 경우에도, $\sum_{n=0}^{\infty} a_n$이 발산한다면 (c_n)은 수렴한다.

5.3.1 연분수계산과정

$x \in \mathbf{R}$이면 x를 정수부분과 1보다 작은 양의 소수부분으로 쓸 수 있다. 즉, $a_0 \in \mathbf{Z}$이고 $0 \le t_0 < 1$일 때

$$x = a_0 + t_0.$$

a_0를 x의 **정수부분(floor)**이라고 부르고, $a_0 = \lfloor x \rfloor$로 표현하기도 한다. $t_0 \neq 0$이면, $\frac{1}{t_0} > 1$이므로

$$\frac{1}{t_0} = a_1 + t_1$$

으로 쓸 수 있는데, 이 때 $a_1 \in \mathbf{N}$이고 $0 \le t_1 < 1$을 만족! 따라서 $t_0 = \frac{1}{a_1+t_1} = [0, a_1 + t_1]$로 쓸 수 있고, 이는 t_0의 (단순연분수일 필요는 없는) 연분수이다. 이 방법을 $t_n \neq 0$인한 계속하면,

$$\frac{1}{t_n} = a_{n+1} + t_{n+1}(\ a_{n+1} \in \mathbf{N},\ 0 \le t_{n+1} < 1)$$

으로 쓸 수 있다. 위와 같은 방법으로 실수 x에 정수들의 수열 $a_0, a_1, a_2, \ldots$를 짝지우는 과정을 **연분수계산과정(continued fraction process)**이라고 부른다.

예 5.3.1. $x = \frac{8}{3}$에 연분수과정을 적용하자. 그러면 $x = 2 + \frac{2}{3}$이므로 $a_0 = 2$, $t_0 = \frac{2}{3}$이다. 또 $\frac{1}{t_0} = \frac{3}{2} = 1 + \frac{1}{2}$이므로, $a_1 = 1$, $t_1 = \frac{1}{2}$이다. 그러면 $\frac{1}{t_1} = 2$이므로, $a_2 = 2$, $t_2 = 0$이 되어 이 과정은 끝난다. 이 연분수과정으로부터 $\frac{8}{3}$의 연분수

$$\frac{8}{3} = [2, 1, 2]$$

를 얻는다.

예 5.3.2. $x = \frac{1+\sqrt{5}}{2}$이면

$$x = 1 + \frac{-1 + \sqrt{5}}{2}$$

이므로 $a_0 = 1$ and $t_0 = \frac{-1+\sqrt{5}}{2}$이다. 그러므로

$$\frac{1}{t_0} = \frac{2}{-1+\sqrt{5}} = \frac{-2-2\sqrt{5}}{-4} = \frac{1+\sqrt{5}}{2},$$

가 되므로 $a_1 = 1$, $t_1 = \frac{-1+\sqrt{5}}{2}$를 얻는다. 비슷한 방법으로 모든 n에 대해서 $a_n = 1$을 얻어 아래의 멋진 식을 얻는다. 물론 등호가 성리함은 (만약 의심된다면) 곧 보이게 된다.

$$\frac{1+\sqrt{5}}{2} = 1 + \cfrac{1}{1 + \cfrac{1}{1 + \cfrac{1}{1 + \cfrac{1}{1 + \cfrac{1}{1+\cdots}}}}}$$

SAGE 예 5.3.3. 예 5.3.2의 등호는 아래 Sage계산과 일치한다.

```
sage: def cf(bits):
...   x = (1 + sqrt(RealField(bits)(5))) / 2
...   return continued_fraction(x)
sage: cf(10)
[1, 1, 1, 1, 1, 1, 1]
sage: cf(30)
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
 1, 1, 1]
sage: cf(50)
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
```

예 5.3.4. $x = e = 2.71828182\ldots$이라고 하자. 연분수과정을 사용하면,

$$a_0, a_1, a_2, \ldots = 2, 1, 2, 1, 1, 4, 1, 1, 6, 1, 1, 8, 1, 1, 10, \ldots$$

를 얻는다. 예를 들어 $a_0 = 2$는 e의 정수부분이다. 2를 뺀 후 뒤집으면(즉, 곱셈역수를 만들면), $1/0.718\ldots = 1.3922\ldots$이 되므로 $a_1 = 1$이다. 1을 뺀 후 또 곱셈역수를 만들면 $1/0.3922\ldots = 2.5496\ldots$이므로 $a_2 = 2$이다. 5.4절에서 e는 패턴이 간단한 연분수를 가짐을 보이려고 한다.

e의 연분수의 다섯 번째 부분수렴은

$$[a_0, a_1, a_2, a_3, a_4, a_5] = \frac{87}{32} = 2.71875$$

인데,

$$\left|\frac{87}{32} - e\right| = 0.000468\ldots.$$

인 의미에서 e의 좋은 유리수 근사값이다. $0.000468\ldots < 1/32^2 = 0.000976\ldots$임을 확인할 수 있는데, 이는 따름정리 5.3.11의 결과를 보여주는 한 예이다.

$\pi = 3.14159265358979\ldots$에도 같은 연분수과정을 적용해 보자. 그러면 π의 연분수는

$$a_0, a_1, a_2, \ldots = 3, 7, 15, 1, 292, 1, 1, 1, 2, 1, 3, 1, 14, \ldots$$

이다. 처음 몇 개의 부분수렴들은 다음과 같다.

$$3, \frac{22}{7}, \frac{333}{106}, \frac{355}{113}, \frac{103993}{33102}, \ldots$$

물론 이들도 π의 좋은 유리수 근사값들인데, 예를 들어,

$$\frac{103993}{33102} = 3.14159265301\ldots$$

이 성립한다. e의 연분수는 (5.4절 참조) 멋진 패턴을 나타내지만 π의 연분수에는 저자가 파악할 수 있는 어떠한 패턴도 보이지 않는다. π의 연분수는 아주 광법위하게 연구되었고, 2천만개 이상의 항들이 계산되었다. 자료는 모든 정수가 π의 연분수과정에서 무한히 자주 나타나고 있음을 제시하고 있다. π나 혹은 이 책에 소개한 어떤 수열의 연분수에 대해 더 알고 싶으면 [50]안에서 각 수열의 처음 몇 항들을 타이핑해 보기 바란다.

5.3.2 무한연분수의 수렴

보조정리 5.3.5. *앞의 연분수과정에서 a_n이 정의되는 모든 n에 대하여,*

$$x = [a_0, a_1, \ldots, a_n + t_n]$$

가 성립한다. 또, $t_n \neq 0$이면 $x = [a_0, a_1, \ldots, a_n, \frac{1}{t_n}]$이다.

증명 귀납법을 사용한다. $x = a_0 + t_0 = a_0 + 1/(1/t_0)$이므로 $n = 0$이면 두 명제가 성립한다. 두번째 명제가 $n-1$에 대해서 성립한다면,

$$\begin{aligned} x &= \left[a_0, a_1, \ldots, a_{n-1}, \frac{1}{t_{n-1}}\right] \\ &= [a_0, a_1, \ldots, a_{n-1}, a_n + t_n] \\ &= \left[a_0, a_1, \ldots, a_{n-1}, a_n, \frac{1}{t_n}\right]. \end{aligned}$$

위 식의 두 번째와 마지막 줄이 각각 n일 때의 보조정리의 첫 번째 명제와 두 번째 명제가 성립함을 보여준다. □

정리 5.3.6 (연분수 극한)**.** *$a_0, a_1, \ldots$은 정수의 수열이고, $n \geq 1$이면 $a_n > 0$이다. $n \geq 0$인 각 n에 대하여, $c_n = [a_0, a_1, \ldots a_n]$이라고 하면, $\lim_{n\to\infty} c_n$이 존재한다.*

증명 정수 m보다 작은 모든 n에 대해서 c_n은 $[a_0, \ldots, a_m]$의 부분수렴이다. 따라서 기초정리 5.2.13에 의해서, 짝수번째 부분수렴 c_{2n}들은 단조증가하고, 홀수번째 부분수렴 c_{2n+1}들은 단조감소한다. 그뿐만 아니라 단조증가하는 짝수번째 부분수렴값들은 모두 c_1보다 작거나 같고, 단조감소하는 홀수번째 부분수렴값들은 모두 c_0보다 크거나 같다. 따라서 $\alpha_0 = \lim_{n\to\infty} c_{2n}$와 $\alpha_1 = \lim_{n\to\infty} c_{2n+1}$가 모두 존재하고, $\alpha_0 \leq \alpha_1$이다. 기초정리 5.2.7에 의해서

$$|c_{2n} - c_{2n-1}| = \frac{1}{q_{2n} \cdot q_{2n-1}} \leq \frac{1}{2n(2n-1)} \to 0$$

이므로, $\alpha_0 = \alpha_1$가 성립한다. □

이제 무한연분수의 값을

$$[a_0, a_1, \ldots] = \lim_{n\to\infty} c_n.$$

으로 정의한다.

예 5.3.7. $x = \pi$로 가지고 위의 정리를 확인해보자. 정리 5.3.6의 증명에서처럼, c_n은 π의 n번 째 부분수렴이라고 하자. n이 홀수들일 때

$$c_1 = 3.1428571\ldots,\ c_3 = 3.1415929\ldots,\ c_5 = 3.1415926\ldots$$

등으로 c_n은 π에 아래로 수렴한다, 반면에 n 이 짝수이면

$$c_2 = 3.1415094\ldots,\ c_4 = 3.1415926\ldots,\ c_6 = 3.1415926\ldots$$

과 같이 값이 점차 커지면서 π에 수렴한다.

정리 5.3.8. *실수들의 수열 $a_0, a_1, a_2, \ldots$은 $n \geq 1$인 모든 n에 대해서 $a_n > 0$이다. 0보다 큰 모든 정수 n에 대하여 $c_n = [a_0, a_1, \ldots a_n]$ 이라고 놓자. 그러면 $\lim_{n\to\infty} c_n$이 존재하기 위한 필요충분조건은 $\sum_{n=0}^{\infty} a_n$이 발산하는 것이다.*

증명 여기서는 $\sum a_n$이 발산하면 $\lim_{n\to\infty} c_n$이 존재하는 것만을 보인다. 다른 방향의 증명은 [52, Ch. 2, Thm. 6.1]에서 볼 수 있다.

정수 q_n은 절 5.2.1에서 $q_{-2} = 1$, $q_{-1} = 0$, 그리고 $n \geq 0$이면

$$q_n = a_n q_{n-1} + q_{n-2}$$

으로 정의한 부분수렴의 분모들의 수열이라고 하자. 정리 5.3.6의 증명에서 보았듯이 극한 $\lim_{n\to\infty} c_n$은, $\{q_n q_{n-1}\}$이 무한대로 발산할 때, 수렴한다. 따라서 $\sum_{n=0}^{\infty} a_n$이 발산하면 $\{q_n q_{n-1}\}$이 무한대로 발산함을 보이면 충분하다.

짝수 n에 대하여

$$\begin{aligned} q_n &= a_n q_{n-1} + q_{n-2} \\ &= a_n q_{n-1} + a_{n-2} q_{n-3} + q_{n-4} \\ &= a_n q_{n-1} + a_{n-2} q_{n-3} + a_{n-4} q_{n-5} + q_{n-6} \\ &= a_n q_{n-1} + a_{n-2} q_{n-3} + \cdots + a_2 q_1 + q_0 \end{aligned}$$

이고, 홀수 n에 대해서는,

$$q_n = a_n q_{n-1} + a_{n-2} q_{n-3} + \cdots + a_1 q_0 + q_{-1}$$

이다. $n > 0$이면 $a_n > 0$이므로 $\{q_n\}$은 증가수열이다. 따라서 $i \geq 0$이면 $q_i \geq 1$이다. 이 사실을 q_n에 대한 위의 식에 적용하면, 짝수인 n은

$$q_n \geq a_n + a_{n-2} + \cdots + a_2$$

을, 홀수인 n은

$$q_n \geq a_n + a_{n-2} + \cdots + a_1$$

을 만족한다.

만약 $\sum a_n$이 발산하면 $\sum a_{2n}$나 $\sum a_{2n+1}$ 중에 적어도 하나는 발산한다. 그러면 위의 부등식들은 $\{q_{2n}\}$이나 $\{q_{2n+1}\}$중 적어도 하나는 발산해야한다. 그런데 $\{q_n\}$은 증가수열이므로, $\{q_n q_{n-1}\}$이 무한대로 발산한다. □

예 5.3.9. $n \geq 2$일 때 $a_n = \frac{1}{n \log(n)}$이라 하자. 적분판정법에 의하여, $\sum a_n$은 발산한다. 따라서 정리 5.3.8에 의하여, 연분수 $[a_0, a_1, a_2, \ldots]$은 수렴한다. 이 연분수는 아주 느리게 수렴한다. 예를 들어, 다음 값들을 참고하면 10000항까지 계산으로도 이들이 수렴할 것인지를 확신하기는 어렵다.

$$[a_0, a_1, \ldots, a_{9999}] = 0.5750039671012225425930\ldots$$

이나

$$[a_0, a_1, \ldots, a_{10000}] = 0.7169153932917378550424\ldots.$$

정리 5.3.10. *실수 x는 연분수과정에 의하여 얻은 (유한, 혹은 무한)단순연분수 $[a_0, a_1, a_2, \ldots]$ 의 값과 같다.*

증명 만약 유한하면 적당한 n에서 $t_n = 0$이 되고, 이는 바로 보조정리 5.3.5의 결과이다. 무한 단순연분수라면, 보조정리 5.3.5에 의하여,

$$x = [a_0, a_1, \ldots, a_n, \frac{1}{t_n}]$$

으로 쓸 수 있고, 기초정리 5.2.5에 의하여 다음을 얻는다.

$$x = \frac{\frac{1}{t_n} \cdot p_n + p_{n-1}}{\frac{1}{t_n} \cdot q_n + q_{n-1}}.$$

따라서, 만약 $c_n = [a_0, a_1, \ldots, a_n]$이라면,

$$\begin{aligned} x - c_n &= x - \frac{p_n}{q_n} \\ &= \frac{\frac{1}{t_n}p_nq_n + p_{n-1}q_n - \frac{1}{t_n}p_nq_n - p_nq_{n-1}}{q_n\left(\frac{1}{t_n}q_n + q_{n-1}\right)}. \\ &= \frac{p_{n-1}q_n - p_nq_{n-1}}{q_n\left(\frac{1}{t_n}q_n + q_{n-1}\right)} \\ &= \frac{(-1)^n}{q_n\left(\frac{1}{t_n}q_n + q_{n-1}\right)} \end{aligned}$$

이 되므로,

$$\begin{aligned} |x - c_n| &= \frac{1}{q_n\left(\frac{1}{t_n}q_n + q_{n-1}\right)} \\ &< \frac{1}{q_n(a_{n+1}q_n + q_{n-1})} \\ &= \frac{1}{q_n \cdot q_{n+1}} \leq \frac{1}{n(n+1)} \to 0 \end{aligned}$$

를 얻는다. 위의 부등식에서는 a_{n+1}이 $\frac{1}{t_n}$의 정수부분이고, 따라서 $t_n < 1$ 이므로 $1 < \frac{1}{t_n}$임을 이용하였다. □

다음 따름정리는 정리 5.3.10의 증명과정에서 확인한 것이다.

따름정리 5.3.11 (연분수의 수렴)**.** *$a_0, a_1, \ldots$이 단수연분수를 정의하고 $x = [a_0, a_1, \ldots] \in \mathbf{R}$이라고 하자. 그러면 모든 m에 대하여,*

$$\left| x - \frac{p_m}{q_m} \right| < \frac{1}{q_m \cdot q_{m+1}}$$

이 성립한다.

기초정리 5.3.12. *x가 유리수이면, 연분수 과정에서 얻은 수열 $a_0, a_1, \ldots$ 은 유한수열이다.*

증명 $[b_0, b_1, \ldots, b_m]$는 알고리듬 1.1.13을 이용하여 얻은 유리수 x의 연분수라고 하면, b_i는 각 단계에서의 몫이다. $m = 0$이면, x는 정수이므로, $m > 0$이라고 가정하자. 그러면 i

$$x = b_0 + 1/[b_1, \ldots, b_m]$$

으로 쓸 수 있다. 만약 $[b_1, \ldots, b_m] = 1$이면, $m = 1$, $b_1 = 1$이 되는데, 이는 정수 $b_0 + 1$의 연분수 $[b_0 + 1]$를 주는 것이므로, 우리의 제한 조건하에서는 알고리듬 1.1.13으로부터 나타나지 않는다. 따라서 $[b_1, \ldots, b_m] > 1$이 되고, 연분수 알고리듬에서는 $a_0 = b_0$, $t_0 = 1/[b_1, \ldots, b_m]$을 얻는다. 이 과정을 반복하면, 모든 n에 대하여 $a_n = b_n$을 얻는다. □

5.4 e의 연분수

e의 연분수는 $[2, 1, 2, 1, 1, 4, 1, 1, 6, \ldots]$로 시작한다. 오일러가 1737년 증명하였듯이([17]참조), e는 분명한 패턴을 가지는 무한연분수로 나타난다. 오일러는 e가 무한연분수로 나타나므로 e가 무리수임을 증명하였다.

이 책에서의 증명은 [9]에 있는 증명을 거의 그대로 설명하고 있다. [9]의 증명 또한 Hermite의 증명을 약간 변형한 것이다 ([40]참조). e의 연분수 표현은 독일책 [41]에서도 다루고 있지만, 그 증명은 이 책에서 다루지 않은 배경을 상당히 많이 이용한다.

5.4.1 준비

먼저 e의 연분수를 약간 다른 형태로 써 보려고 한다. $[2, 1, 2, 1, 1, 4, \ldots]$ 대신 전체적으로 같은 패턴을 가지도록 하기 위하여 수열 $\{a_n | n \geq 0\}$을

$$[1, 0, 1, 1, 2, 1, 1, 4, \ldots, 2(n-1), 1, 1, \ldots,]$$

로 놓고 시작할 수 있다. (이 단원의 다른 절에서는 $n \geq 1$이면 부분 몫 a_n은 양수로 가정하지만, 이 절에서는 잠시 이 조건을 잊고 $a_1 = 0$을 허락해준다.) 세 수열에 의해 주어진 연분수의 부분수렴의 분자와 분모는 간단한 순환성질을 만족한다. p_i나 q_i 대신 r_i를 사용하면 다음과 같다.

$$\begin{aligned} r_{3n} &= r_{3n-1} + r_{3n-2} \\ r_{3n-1} &= r_{3n-2} + r_{3n-3} \\ r_{3n-2} &= 2(n-1)r_{3n-3} + r_{3n-4}. \end{aligned}$$

우리의 첫 번째 목표는 위의 세 개의 점화식을 r_{3n}, r_{3n-3}, r_{3n-6}만 포함하는 한 개의 점화식으로 줄이는 것이다. 먼저 위의 세 식으로부터 다음 식을

TABLE 5.1. Convergents

n	0	1	2	3	4	$\cdots$
x_n	1	3	19	193	2721	$\cdots$
y_n	1	1	7	71	1001	$\cdots$
x_n/y_n	1	3	$2.714\ldots$	$2.71830\ldots$	$2.7182817\ldots$	$\cdots$

얻는다.

$$\begin{aligned} r_{3n} &= r_{3n-1} + r_{3n-2} \\ &= (r_{3n-2} + r_{3n-3}) + (2(n-1)r_{3n-3} + r_{3n-4}) \\ &= (2(n-1)r_{3n-3} + r_{3n-4} + r_{3n-3}) + (2(n-1)r_{3n-3} + r_{3n-4}) \\ &= (4n-3)r_{3n-3} + 2r_{3n-4}. \end{aligned}$$

따라서

$$r_{3n-3} = 2r_{3n-7} + (4n-7)r_{3n-6}.$$

첫 번째 식에서 $2r_{3n-4}$를 소거하기 위하여 $2r_{3n-4}$를 다시 써보면 다음과 같다.

$$\begin{aligned} 2r_{3n-4} &= 2(r_{3n-5} + r_{3n-6}) \\ &= 2((2(n-2)r_{3n-6} + r_{3n-7}) + r_{3n-6}) \\ &= (4n-6)r_{3n-6} + 2r_{3n-7} \end{aligned}$$

이 식을 첫 번째 식의 $2r_{3n-4}$에, 그리고 두 번째 식에서 $2r_{3n-7}$를 첫 번째 식에 대입하면 우리가 필요로 하는 점화식

$$r_{3n} = 2(2n-1)r_{3n-3} + r_{3n-6}$$

을 얻는다.

5.4.2 두 적분 수열

$x_n = p_{3n}$, $y_n = q_{3n}$으로 정의하자. 수렴하는 수열의 $3n$-항들의 수렴값은 n-항들의 수렴값과 같으므로, x_n/y_n도 연분수의 극한으로 수렴한다. 수열 $\{x_n\}$, $\{y_n\}$은 $n \geq 2$인 모든 n에 대하여 앞 절에서 얻은 점화식

$$z_n = 2(2n-1)z_{n-1} + z_{n-2} \quad (n \geq 2) \tag{5.4.1}$$

을 만족한다. (단, z_n은 x_n이나 y_n을 대신한다.) 이 두 수열은 표 5.1을 참고한다. (첫 두 항 $x_0 = 1$, $x_1 = 3$, $y_0 = y_1 = 1$의 값들은 원래의 연분수로부터 직접 계산으로 얻은 값들이다.) 그런데 각 단계에서 여러 항들을 뺐으므로, x_n/y_n은 e에 아주 빠르게 수렴한다.

5.4.3 관련 적분 수열들

수열 $T_0, T_1, T_2, \ldots$ 은 $n \geq 0$에 대하여

$$T_n = \int_0^1 \frac{t^n(t-1)^n}{n!}\ e^t dt$$

으로 정의된 실수들의 수열이다. T_0, T_1의 값은 다음과 같다.

$$\begin{aligned}
T_0 &= \int_0^1 e^t dt = e - 1, \\
T_1 &= \int_0^1 t(t-1)e^t dt \\
&= -\int_0^1 ((t-1)+t)e^t dt \\
&= -(t-1)e^t\Big|_0^1 - te^t\Big|_0^1 + 2\int_0^1 e^t dt \\
&= -1 - e + 2(e-1) = e - 3
\end{aligned}$$

(T_1은 $u = t(t-1)$, $dv = e^t dt$으로 놓고 부분적분을 사용한다. 그러면 정적분의 범위가 $[0,1]$구간인데 u가 경계값 0과 1에서 0이 되어 계산이 간단해지는데 이는 $n \geq 1$인 T_n의 계산에서도 적용된다.) 놀랍게도 $T_0 = y_0 e - x_0$, $T_1 = y_1 e - x_1$임을 확인할 수 있다. 이제 T_n이 x_i와 y_i와 같은 점화식 (5.4.1)을 만족한다면, 귀납법에 의하여 $T_n = y_n e - x_n$이 성립함을 쉽게 보일 수 있다. 다음은 T_n도 x_i, y_i와 같은 점화식을 만족함을 보이는 식이다.

$$\begin{aligned}
T_n &= \int_0^1 \frac{t^n(t-1)^n}{n!}\ e^t dt \\
&= -\int_0^1 \frac{t^{n-1}(t-1)^n + t^n(t-1)^{n-1}}{(n-1)!}\ e^t dt \\
&= \int_0^1 \Big(\frac{t^{n-2}(t-1)^n}{(n-2)!} + n\frac{t^{n-1}(t-1)^{n-1}}{(n-1)!} \\
&\qquad\qquad + n\frac{t^{n-1}(t-1)^{n-1}}{(n-1)!} + \frac{t^n(t-1)^{n-2}}{(n-2)!} \Big) e^t dt \\
&= 2nT_{n-1} + \int_0^1 \frac{t^{n-2}(t-1)^{n-2}}{(n-2)!}(2t^2 - 2t + 1)\ e^t dt \\
&= 2nT_{n-1} + 2\int_0^1 \frac{t^{n-1}(t-1)^{n-1}}{(n-2)!}\ e^t dt + \int_0^1 \frac{t^{n-2}(t-1)^{n-2}}{(n-2)!}\ e^t dt \\
&= 2nT_{n-1} + 2(n-1)T_{n-1} + T_{n-2} \\
&= 2(2n-1)T_{n-1} + T_{n-2}.
\end{aligned}$$

그러므로, $T_n = y_n e - x_n$이다. 한편

$$\lim_{n\to\infty} \int_0^1 \frac{t^n(t-1)^n}{n!}\ e^t dt = 0,$$

이므로

$$\lim_{n\to\infty} \frac{x_n}{y_n} = \lim_{n\to\infty} (e - \frac{T_n}{y_n}) = e.$$

이다. 따라서 x_n/y_n이 e에 점근하고, 또 연분수 $[2,1,2,1,1,4,1,1,\ldots]$는 e에 수렴한다.

5.4.4 증명의 확장

이 절의 증명방법을 일반화하면 모든 자연수 $n \in \mathbf{N}$에 대하여 $e^{1/n}$의 연분수가

$$[1,\ (n-1),\ 1,\ 1,\ (3n-1),\ 1,\ 1,\ (5n-1),\ 1,\ 1,\ (7n-1),\ldots]$$

임을 보일 수 있다(Exercise 5.6 참조).

5.5 이차 무리수

이 절에서는 순환무한연분수는 이차무리수임을 보인다. 이차무리수란 이차방정식의 무리수 해를 의미한다. 연분수에서의 이 결과는 소수가 순환소수이기 위한 필요충분조건이 유리수라는 사실과 비교된다. 이차무리수의 연분수가 결국은 순환한다는 사실의 증명은 놀라울 정도로 어렵고 흥미로운 유한성 논증(finiteness argument)을 포함한다. 절 5.5.2에서는 의도적으로 우리가 관심을 보이지 않고 있던 $\mathbf{Q}$상에서의 3차 이상의 기약 다항식의 실근들의 연분수를 강조한다.

정의 5.5.1 (이차 무리수)**. 이차 무리수(quadratic irrational)**는 계수가 유리수인 이차 다항식의 무리수해를 의미한다.

예를 들어 $(1+\sqrt{5})/2$는 이차무리수이다.

$$\frac{1+\sqrt{5}}{2} = [1,1,1,\ldots]$$

였음을 기억하자. $\sqrt{2}$ 의 연분수는 $[1,2,2,2,2,2,\ldots]$, $\sqrt{389}$의 연분수는

$$[19,1,2,1,1,1,1,2,1,38,1,2,1,1,1,1,2,1,38,\ldots]$$

이다. $[1,2,1,1,1,1,2,1,38]$는 영원히 계속될까?

SAGE 예 5.5.2. Sage를 이용하여 $\sqrt{389}$의 항을 더 계산하면 다음과 같다.

```
sage: def cf_sqrt_d(d, bits):
...   x = sqrt(RealField(bits)(d))
...   return continued_fraction(x)
sage: cf_sqrt_d(389,50)
[19, 1, 2, 1, 1, 1, 1, 2, 1, 38, 1, 2, 1, 1, 1, 1, 2, 1, 38, 2]
sage: cf_sqrt_d(389,100)
[19, 1, 2, 1, 1, 1, 1, 2, 1, 38, 1, 2, 1, 1, 1, 1, 2, 1, 38,
 1, 2, 1, 1, 1, 1, 2, 1, 38, 1, 2, 1, 1, 1, 1, 2, 1, 38, 1,
 2, 1, 1]
```

5.5.1 순환연분수

정의 5.5.3 (순환연분수)**. 순환연분수(periodic continued fraction)**는 충분히 큰 모든 n에 대해서

$$a_n = a_{n+h}$$

을 만족하는 정수 h가 존재하는 연분수 $[a_0, a_1, \ldots, a_n, \ldots]$이다. 이런 h 중 가장 작은 정수를 **연분수의 주기(period of the continued fraction)**라고 한다.

예 5.5.4. 순환연분수 $[1, 2, 1, 2, \ldots] = \overline{[1, 2]}$는 어떤 값에 수렴할까?

$$\overline{[1,2]} = 1 + \cfrac{1}{2 + \cfrac{1}{1 + \cfrac{1}{2 + \cfrac{1}{1 + \cdots}}}},$$

이므로, $\alpha = \overline{[1,2]}$라 놓으면,

$$\alpha = 1 + \frac{1}{2 + \dfrac{1}{\alpha}} = 1 + \frac{1}{\dfrac{2\alpha + 1}{\alpha}} = 1 + \frac{\alpha}{2\alpha + 1} = \frac{3\alpha + 1}{2\alpha + 1}$$

가 성립한다. 따라서 $2\alpha^2 - 2\alpha - 1 = 0$이므로

$$\alpha = \frac{1 + \sqrt{3}}{2}$$

이다.

정리 5.5.5 (순환연분수의 특징)**.** *무한단순연분수가 순환연분수이기위한 필요충분조건은 이차무리수를 나타내는 연분수이다.*

증명 ($\Longrightarrow$) 연분수

$$[a_0, a_1, \ldots, a_n, \overline{a_{n+1}, \ldots, a_{n+h}}]$$

이 순환한다고 하자. 그러면 $\alpha = [a_{n+1}, a_{n+2}, \ldots]$라 놓으면

$$\alpha = [a_{n+1}, \ldots, a_{n+h}, \alpha]$$

가 성립하므로, 기초정리 5.2.5에 의하여

$$\alpha = \frac{\alpha p_{n+h} + p_{n+h-1}}{\alpha q_{n+h} + q_{n+h-1}}.$$

따라서 α는 계수가 유리수인 이차식의 해로써 나타난다. 그런데

$$\begin{aligned}[a_0, a_1, \ldots] &= [a_0, a_1, \ldots, a_n, \alpha] \\ &= a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cdots + \cfrac{1}{\alpha}}}\end{aligned}$$

이고 a_i는 모두 정수이므로, 분모의 유리화를 유한 번 반복하면 위 식은 $c + d\alpha$ 또는 $c + d\overline{\alpha}$ ($c, d \in \mathbf{Q}, \overline{\alpha} : \alpha$ 의 켤레무리수) 꼴이 되므로, $[a_0, a_1, \ldots]$는 $\mathbf{Q}$ 위에서 정의된 이차 다항식의 해이다.

기초정리 5.3.12로부터 무한연분수가 수렴하는 값은 유리수는 아니므로 $\alpha \notin \mathbf{Q}$이다.

($\Longleftarrow$) $\alpha \in \mathbf{R}$가 이차식

$$a\alpha^2 + b\alpha + c = 0 \tag{5.5.1}$$

를 만족하는 무리수라고 하자. 단, $a, b, c \in \mathbf{Z}$이고 $a \neq 0$이다. $[a_0, a_1, \ldots]$이 α의 연분수일 때,

$$r_n = [a_n, a_{n+1}, \ldots],$$

라고 놓으면,

$$\alpha = [a_0, a_1, \ldots, a_{n-1}, r_n]$$

을 만족한다. 이제 r_n들로 나타날 수 있는 수들이 유한개뿐임을 보이면 α의 연분수는 순환한다. 왜냐하면, r_n들로 나타날 수 있는 수들이 유한개뿐이면, $r_n = r_{n+h}$를 만족하는 $n, h > 0$가 존재한다. 따라서

$$\begin{aligned}[a_0, \ldots, a_{n-1}, r_n] &= [a_0, \ldots, a_{n-1}, a_n, \ldots, a_{n+h-1}, r_{n+h}] \\ &= [a_0, \ldots, a_{n-1}, a_n, \ldots, a_{n+h-1}, r_n] \\ &= [a_0, \ldots, a_{n-1}, a_n, \ldots, a_{n+h-1}, a_n, \ldots, a_{n+h-1}, r_{n+h}] \\ &= [a_0, \ldots, a_{n-1}, \overline{a_n, \ldots, a_{n+h-1}}]\end{aligned}$$

를 얻는다.

따라서 r_n들로 나타날 수 있는 수들이 유한 개 뿐이라는 것만 보이면 된다.

$$\alpha = \frac{p_n}{q_n} = \frac{r_n p_{n-1} + p_{n-2}}{r_n q_{n-1} + q_{n-2}}$$

이므로, 이를 이차식 (5.5.1)에 대입하면 r_n은 이차식

$$A_n r_n^2 + B_n r_n + C_n = 0$$

을 만족한다. 이 때 $A_n,\ B_n,\ C_n$은 다음과 같다:

$$\begin{aligned}
A_n &= ap_{n-1}^2 + bp_{n-1}q_{n-1} + cq_{n-1}^2,\\
B_n &= 2ap_{n-1}p_{n-2} + b(p_{n-1}q_{n-2} + p_{n-2}q_{n-1}) + 2cq_{n-1}q_{n-2},\\
C_n &= ap_{n-2}^2 + bp_{n-2}q_{n-2} + cq_{n-2}^2.
\end{aligned}$$

위 식에서 $A_n, B_n, C_n \in \mathbf{Z}$이고, $C_n = A_{n-1}$, 그리고

$$B_n^2 - 4A_nC_n = (b^2 - 4ac)(p_{n-1}q_{n-2} - q_{n-1}p_{n-2})^2 = b^2 - 4ac$$

이다.

정리 5.3.10의 증명으로부터

$$\left|\alpha - \frac{p_{n-1}}{q_{n-1}}\right| < \frac{1}{q_nq_{n-1}}$$

가 성립한다. 통분하면

$$|\alpha q_{n-1} - p_{n-1}| < \frac{1}{q_n} < \frac{1}{q_{n-1}}$$

가 되고, 따라서 ($\alpha q_{n-1} - p_{n-1} = \frac{-\delta}{q_{n-1}}$라 놓으면) $|\delta| < 1$이면서

$$p_{n-1} = \alpha q_{n-1} + \frac{\delta}{q_{n-1}}$$

를 만족한다. A_n에 이 식을 대입하면,

$$\begin{aligned}
A_n &= a\left(\alpha q_{n-1} + \frac{\delta}{q_{n-1}}\right)^2 + b\left(\alpha q_{n-1} + \frac{\delta}{q_{n-1}}\right)q_{n-1} + cq_{n-1}^2\\
&= (a\alpha^2 + b\alpha + c)q_{n-1}^2 + 2a\alpha\delta + a\frac{\delta^2}{q_{n-1}^2} + b\delta\\
&= 2a\alpha\delta + a\frac{\delta^2}{q_{n-1}^2} + b\delta.
\end{aligned}$$

따라서 다음 부등식을 얻는다.

$$|A_n| = \left|2a\alpha\delta + a\frac{\delta^2}{q_{n-1}^2} + b\delta\right| < 2|a\alpha| + |a| + |b|.$$

이는 정수 A_n이 오직 유한개의 값만을 취할 수 있음을 보여준다. 또,

$$|C_n| = |A_{n-1}| \quad , \quad |B_n| = \sqrt{b^2 - 4(ac - A_nC_n)}$$

이므로 세 쌍의 정수 (A_n, B_n, C_n)이 취할 수 있는 값들도 유한개의 가능성 밖에 없으므로, 이들을 계수로 갖는 이차 다항식의 해로 나타나는 r_n도 유한개뿐이고, 따라서 정리가 성립한다. (위의 증명은 [23, Thm. 177, pg.144–145]을 많이 참고하였다.) □

5.5.2 고차 대수적 수의 연분수

정의 5.5.6 (대수적 수)**. 대수적 수(algebraic number)**는 다항식 $f \in \mathbf{Q}[x]$의 해이다.

Open Problem 5.5.7. *대수적 수 $\sqrt[3]{2}$의 연분수를 완벽하게 묘사하여라. 시작은 다음과 같다.*

$$[1, 3, 1, 5, 1, 1, 4, 1, 1, 8, 1, 14, 1, 10, 2, 1, 4, 12, 2, 3, 2, 1, 3, 4, 1, 1, 2, 14,$$
$$3, 12, 1, 15, 3, 1, 4, 534, 1, 1, 5, 1, 1, \ldots]$$

본 저자는 아직은 어떤 특정한 패턴(pattern)을 찾지 못하였고, 534라는 수의 출현으로 풀 수 있으리라는 자신감마저 줄어 들었다. Lang과 Trotter ([35] 참조)는 $\sqrt[3]{2}$의 연분수의 많은 항들을 통계적으로 분석하였는데 그들의 결과는 $\sqrt[3]{2}$는 특이한("unusual") 연분수 형태임을 시사하는데, 그러나 그 후 [36]의 결과는 그렇지 않을 수 있음을 시사한다.

Khintchine ([26, pg. 59])

> [1963년을 기준으로] 고차 대수적 수의 연분수 표현에 대하여는 우리가 지금까지 연분수들에 대해 증명한 사실들과 유사한 어떤 성질들도 알려져 있지 않다. [...] 현재까지 *이차 이외의 고차 대수적 수의 연분수에 대해 알려진 사실이 없다*는 사실을 지적한다는 것이 흥미롭다. 심지어 이 연분수들의 각 항이 유계인지도 모른다. 일반적으로 2차보다 큰 대수적 수의 연분수는 아주 어렵다고 알려져 있고 거의 연구되지 않은 상태이다.

Richard Guy ([19, pg. 260]참조)

> 단순연분수의 부분몫이 유계가 아닌 3차 이상의 대수적 수는 존재하는가? 모든 그런 대수적 수의 부분 몫(partial quotients)은 유계가 아닌가?

Baum과 Sweet는 Richard Guy의 질문에 $\mathbf{Q}$를 다른 체 K위에서의 대수적 수로 바꾼 후 답을 주었다[4]. (체 K는 원소가 두 개인 체 $\mathbf{F}_2$위에서 변수 $1/x$의 Laurent 급수들의 체 $\mathbf{F}_2((1/x))$였다. 즉, K의 원소는 x에 관한 다항식과 $1/x$에 관한 멱급수의 합이다.) 그들은 연분수의 모든 항들의 차수가 유한한 K위에서의 3차 대수적 원소도 찾았고, 또 연분수의 항들의 차수가 유계하지 않은 3차 이상의 여러 대수적 원소들을 찾았다.

5.6 유리수 인식하기

어떤 유리수의 근사값을 어떻게든 구했다고 하고, 그 유리수가 무엇인지 알아내고 싶다고 하자. 아주 정밀하게 그 값을 계산하기 위하여, 소수 전개에서의 주기를 찾는 것은, 주기는 굉장히 큰 수가 될 수 있기 때문에 (아래 참조), 좋은 접근 방법은 아니다. 훨씬 좋은 방법은 그 근사값의 단순연분수를 계산하고, 아주 큰 부분 몫이 나오기 전에 잘라서 그 자른 연분수의 값을 계산하는 것이다. 그 결과 상대적으로 작은 분자와 분모를 갖는 유리수를 얻는데, 이 수는, 연분수의 꼬리 부분은 $1/a_n$을 넘지 않으므로, 우리가 구하고자 하는 유리수와는 상당히 가까운 수이다.

어떻게 유리수를 인식하는지 약간은 부자연스러운 예로 살펴본다. 찾고자 하는 유리수는 x이다.

$$x = 9495/3847 = 2.468157005458799064205874707564335846113 85\ldots.$$

그 근사값을 2.468157005458799064라고 하면, 이 수의 연분수는 다음과 같다.

$$[2, 2, 7, 2, 1, 5, 1, 1, 1, 1, 1, 1, 328210621945, 2, 1, 1, 1, \ldots]$$

그리고 다음 식을 확인할 수 있다.

$$[2, 2, 7, 2, 1, 5, 1, 1, 1, 1, 1, 1] = \frac{9495}{3847}.$$

유리수는 반드시 순환소수가 됨을 알고 있지만 위의 x의 소수 표현에서 어떤 주기도 확인할 수 없었다. 법 3847에서 10의 위수가 3846이므로, $1/3847$은 주기가 3846인 순환소수가 된다(Exercise 5.7 참조).

이 예보다는 덜 인위적인 유리수 값을 찾는 응용을 보기 위하여 $f(x) \in \mathbf{Z}[x]$는 적어도 한 개의 유리수 해를 가지는 계수가 정수인 다항식이라고 가정하자. 그러면 뉴톤의 방법으로 각 실수해의 근사값들을 구할 수 있고, 그 근사값들의 연분수들을 이용하여 유리수 근사값을 구한 후 식에 대입하여 해가 되는지를 확인할 수 있다. 다항식의 유리수 해 n/d는, $n, d \in \mathbf{Z}$를 서로 소로 놓으면, n는 상수항을, d는 최고차항의 계수를 나눈다는 사실도 이용할 수 있다. 그러나 이 정리를 이용하려면 f의 상수항과 최고차항의 계수를 인수분해해야하는데(절 1.1.3 참조), 만약 이 수들이 굉장히 큰 수들인 경우는 이 방법은 실용적이지는 않다. 반면, 뉴턴의 방법과 연분수를 이용하는 방법은 f의 차수가 너무 크지 않으면 빠르게 n/d을 구할 수 밖에 없다.

예를 들어, $f = 3847x^2 - 14808904x + 36527265$라고 하자. 뉴턴의 방법을 적용하기 위하여, x_0를 f의 해라고 가정하자. 다음 식

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}.$$

을 반복하면 x_n이 f의 진짜 해의 근사값이라는 것이 뉴턴의 방법이다. $x_0 = 0$으로 잡자. 그러면

$$x_1 = 2.466574501394566404103909378,$$

$$x_2 = 2.468157004807401923043166846.$$

x_1과 x_2의 연분수는 각각 다음과 같다:

$$[2, 2, 6, 1, 47, 2, 1, 4, 3, 1, 5, 8, 2, 3]$$

$$[2, 2, 7, 2, 1, 5, 1, 1, 1, 1, 1, 1, 103, 8, 1, 2, 3, \ldots].$$

x_2의 연분수를 103 앞에서 자르면,

$$[2, 2, 7, 2, 1, 5, 1, 1, 1, 1, 1, 1]$$

를 얻는데, 계산하면 $9495/3847$이고, f에 대입하여 f의 유리수 해임을 확인할 수 있다.

SAGE 예 5.6.1. SAGE를 이용하여 위의 계산을 한다. 먼저 Newton의 반복을 구현한다.

```
sage: def newton_root(f, iterates=2, x0=0, prec=53):
...     x = RealField(prec)(x0)
...     R = PolynomialRing(ZZ,'x')
...     f = R(f)
...     g = f.derivative()
...     for i in range(iterates):
...         x = x - f(x)/g(x)
...     return x
```

다음 Newton의 반복을 run 하고, 그 나온 값의 연분수를 계산한다.

```
sage: a = newton_root(3847*x^2 - 14808904*x + 36527265); a
2.46815700480740
sage: cf = continued_fraction(a); cf
[2, 2, 7, 2, 1, 5, 1, 1, 1, 1, 1, 1, 103, 8, 1, 2, 3]
```

연분수를 잘라 내고 그 값을 계산한다.

```
sage: c = cf[:12]; c
[2, 2, 7, 2, 1, 5, 1, 1, 1, 1, 1, 1]
sage: c.value()
9495/3847
```

이 책의 범위를 넘어서긴 하지만 연분수의 또 다른 종류의 중요한 응용을 소개하면, 고급 정수론의 어떤 영역에는 어떤 점들에 유리수값들을 갖는 함수들이 있고 오직 근사값들을 이용하여 값을 구해야 할 때, 위에서 보여준 연분수 방법이 이 함수값을 구하는데 결정적 역할을 한다.

5.7 두 제곱수의 합

이 절에서는 연분수를 응용하여 다음 정리를 증명한다.

정리 5.7.1. *양의 정수 n이 두 제곱수의 합이 되기 위한 필요충분조건은 n의 소인수분해에서 $p \equiv 3 \pmod 4$을 만족하는 n의 소수 약수 p의 지수가 짝수인 것이다.*

먼저 몇 개의 예들을 살펴본다. $5 = 1^2 + 2^2$는 두 제곱수의 합이지만, 7은 그렇지 않다. 2001은 3으로 나누어지나 9로 나누어지지 않으므로 정리 5.7.1로부터 2001은 두 제곱수의 합이 아니다 (왜냐하면 $2+1$은 3으로 나누어지나 $2+1$은 9로 나누어지지 않으므로). 이 정리는 $2 \cdot 3^4 \cdot 5 \cdot 7^2 \cdot 13$는 두 제곱수의 합임을 말해준다.

SAGE 예 5.7.2. Sage을 사용하여 주어진 정수 n이 두 제곱수의 합인지 아닌지를 그대로 알려주고, 또 그런 경우 $a^2+b^2 = n$이 되는 정수 a, b를 계산해주는 프로그램을 설계한다.

```
sage: def sum_of_two_squares_naive(n):
...     for i in range(int(sqrt(n))):
...         if is_square(n - i^2):
...             return i, (Integer(n-i^2)).sqrt()
...     return "%s is not a sum of two squares"%n
```

다음 두 세 경우에 우리 함수를 사용한다.

```
sage: sum_of_two_squares_naive(23)
'23 is not a sum of two squares'
```

```
sage: sum_of_two_squares_naive(389)
(10, 17)
sage: sum_of_two_squares_naive(2007)
'2007 is not a sum of two squares'
sage: sum_of_two_squares_naive(2008)
'2008 is not a sum of two squares'
sage: sum_of_two_squares_naive(2009)
(28, 35)
sage: 28^2 + 35^2
2009
sage: sum_of_two_squares_naive(2*3^4*5*7^2*13)
(189, 693)
```

정의 5.7.3 (Primitive)**.** 정수 n이 두 제곱수의 합 $n = x^2 + y^2$에서 x와 y가 서로 소이면서 **primitive** 표현이라고 한다.

보조정리 5.7.4. *정수 n이 $p \equiv 3 \pmod 4$인 소수 p로 나누어지면, n은 primitive 하게 두 제곱수의 합으로 표현될 수 없다.*

증명 정수 n이 x와 y가 서로 소이면서 $n = x^2 + y^2$이고, p는 n의 임의의 소수 약수라고 하자. 그러면

$$p \mid x^2 + y^2, \quad \gcd(x, y) = 1$$

이므로, $p \nmid x$와 $p \nmid y$를 만족한다. 그런데 $\mathbf{Z}/p\mathbf{Z}$는 체이므로, 식 $x^2 + y^2 \equiv 0 \pmod p$를 y^2로 나눌 수 있고, 그러면 $(x/y)^2 \equiv -1 \pmod p$이 성립한다. 따라서 -1이 법 p에서 제곱수가 되므로 Legendre 부호 $\left(\frac{-1}{p}\right)$는 $+1$이 되어야 한다. 그런데 기초정리 4.2.1에 의하면

$$\left(\frac{-1}{p}\right) = (-1)^{(p-1)/2}$$

인데 $\left(\frac{-1}{p}\right) = 1$은 $(p-1)/2$이 짝수, 즉, $p \equiv 1 \pmod 4$와 동치이다. □

정리 5.7.1의 증명 ($\Longrightarrow$)**.** p는 $p \equiv 3 \pmod 4$를 만족하는 소수, r은 $p^r \mid n$이나 $p^{r+1} \nmid n$인 홀수이고, $n = x^2 + y^2$이라고 가정하자. $d = \gcd(x, y)$라고 하면,

$$x = dx', \quad y = dy', \quad \text{그리고} \quad n = d^2 n'$$

이면서 $\gcd(x', y') = 1$,

$$(x')^2 + (y')^2 = n'$$

이 된다. 즉, n'은 서로 소인 두 제곱수의 합이 된다. r이 홀수이므로, p^r이 모두 d^2를 나눌 수가 없으므로, $p \mid n'$이어야만 한다. 그러면 보조정리 5.7.4에 의하여 $\gcd(x', y') > 1$이어야만 하므로, n이 두 제곱수의 합으로 표현된다는 우리의 가정은 모순이다. □

이제 정리 5.7.1의 충분조건 ($\Longleftarrow$)을 증명하기 위하여 이 문제는 n이 소수인 경우만 증명하면 충분함을 보이자. 먼저 n을 $n = n_1^2 n_2$으로 놓자. 이 때 n_2는 $p \equiv 3 \pmod 4$인 소수 약수가 없는 수이다. 그러면

$$(x_1^2 + y_1^2)(x_2^2 + y_2^2) = (x_1x_2 - y_1y_2)^2 + (x_1y_2 + x_2y_1)^2 \tag{5.7.1}$$

이 성립하여, 두 제곱수의 합의 곱은 다시 두 제곱수의 합이므로, n_2의 모든 소수 약수가 두 제곱수의 합임을 보이면 된다. $2 = 1^2 + 1^2$이므로, $p \equiv 1 \pmod 4$인 소수 p가 두 제곱수의 합임을 보이면 충분하다.

보조정리 5.7.5. *$x \in \mathbf{R}$, $n \in \mathbf{N}$이면, 분모가 n이하이면서*

$$\left| x - \frac{a}{b} \right| \leq \frac{1}{b(n+1)}$$

를 만족하는 기약분수 $\dfrac{a}{b}$가 존재한다.

증명 x의 연분수 $[a_0, a_1, \ldots]$를 생각하자. 따름정리 5.3.11에 의하여, 각 m에 대하여

$$\left| x - \frac{p_m}{q_m} \right| < \frac{1}{q_m \cdot q_{m+1}}$$

이 성립한다. $q_{m+1} \geq q_m + 1$이고 $q_0 = 1$이므로, $q_m \leq n < q_{m+1}$을 만족하는 m이 존재하거나, 혹은 x의 연분수가 유한하면서 n이 x의 분모보다 크다. 후자의 경우에는 $\frac{a}{b} = x$로 택하면 된다. 전자의 경우는

$$\left| x - \frac{p_m}{q_m} \right| < \frac{1}{q_m \cdot q_{m+1}} \leq \frac{1}{q_m \cdot (n+1)}$$

이 성립하므로 $\dfrac{a}{b} = \dfrac{p_m}{q_m}$로 택하면 된다. □

정리 5.7.1의 증명 ($\Longleftarrow$)**.** 앞에서 설명한 것처럼 $p \equiv 1 \pmod 4$인 임의의 소수는 두 제곱수의 합임을 보이면 된다. $p \equiv 1 \pmod 4$이므로,

$$(-1)^{(p-1)/2} = 1$$

이다. 따라서 기초정리 4.2.1로부터 -1이 법 p에서 제곱수이다. 즉, $r \in \mathbf{Z}$이 존재하여 $r^2 \equiv -1 \pmod p$을 만족한다. $n = \lfloor \sqrt{p} \rfloor$, $x = -\frac{r}{p}$ 로 놓고 보조정리 5.7.5를 적용하면, 기약분수 $\dfrac{a}{b}$가 존재하여 $0 < b < \sqrt{p}$와

$$\left| -\frac{r}{p} - \frac{a}{b} \right| \leq \frac{1}{b(n+1)} < \frac{1}{b\sqrt{p}}$$

를 만족한다. $c = rb + pa$로 놓으면,

$$|c| < \frac{pb}{b\sqrt{p}} = \frac{p}{\sqrt{p}} = \sqrt{p}$$

이므로

$$0 < b^2 + c^2 < 2p$$

를 얻는다. 그런데 $c \equiv rb \pmod{p}$이므로,

$$b^2 + c^2 \equiv b^2 + r^2 b^2 \equiv b^2(1 + r^2) \equiv 0 \pmod{p}$$

이 되는데 $2p$보다 작은 양수 중에서 p의 배수는 p뿐이므로 $b^2 + c^2 = p$를 얻는다. □

참조 5.7.6. 정리 5.7.1의 증명으로부터 임의의 $p \equiv 1 \pmod{4}$를 두 제곱수의 합으로 표현할 수 있는 효율적인 알고리듬을 얻을 수 있다.

SAGE 예 5.7.7. Sage와 정리 5.7.1를 이용하면 $p \equiv 1 \pmod{4}$인 소수 p를 두 제곱수의 합으로 표현하는 효율적인 알고리듬을 얻는다. 먼저 정리의 증명으로부터 알고리듬을 구현한다.

```
sage: def sum_of_two_squares(p):
...     p = Integer(p)
...     assert p%4 == 1, "p must be 1 modulo 4"
...     r = Mod(-1,p).sqrt().lift()
...     v = continued_fraction(-r/p)
...     n = floor(sqrt(p))
...     for x in v.convergents():
...         c = r*x.denominator() + p*x.numerator()
...         if -n <= c and c <= n:
...             return (abs(x.denominator()),abs(c))
```

다음 예는 이 알고리듬을 이용하여 $\equiv 1 \pmod{4}$인 첫 번째 10 자릿수 소수를 두 제곱수의 합으로 쓴다.

```
sage: p = next_prime(next_prime(10^10))
sage: sum_of_two_squares(p)
(55913, 82908)
```

위의 계산은 본질적으로 순간적인 계산이다. 대신 먼저 설명한 naive 알고리듬을 이용하면 p를 두 제곱수의 합으로 표현하는 데 수 초가 걸린다.

```
sage: sum_of_two_squares_naive(p)
(55913, 82908)
```

5.8 Exercises

5.1 If $c_n = p_n/q_n$ is the nth convergent of $[a_0, a_1, \ldots, a_n]$ and $a_0 > 0$, show that

$$[a_n, a_{n-1}, \ldots, a_1, a_0] = \frac{p_n}{p_{n-1}}$$

and

$$[a_n, a_{n-1}, \ldots, a_2, a_1] = \frac{q_n}{q_{n-1}}.$$

(Hint: In the first case, notice that $\dfrac{p_n}{p_{n-1}} = a_n + \dfrac{p_{n-2}}{p_{n-1}} = a_n + \dfrac{1}{\frac{p_{n-1}}{p_{n-2}}}$.)

5.2 Show that every nonzero rational number can be represented in exactly two ways by a finite simple continued fraction. (For example, 2 can be represented by $[1, 1]$ and $[2]$, and $1/3$ by $[0, 3]$ and $[0, 2, 1]$.)

5.3 Evaluate the infinite continued fraction $[2, \overline{1, 2, 1}]$.

5.4 Determine the infinite continued fraction of $\frac{1+\sqrt{13}}{2}$.

5.5 Let $a_0 \in \mathbf{R}$ and $a_1, \ldots, a_n$ and b be positive real numbers. Prove that

$$[a_0, a_1, \ldots, a_n + b] < [a_0, a_1, \ldots, a_n]$$

if and only if n is odd.

5.6 (*) Extend the method presented in the text to show that the continued fraction expansion of $e^{1/k}$ is

$$[1,\ (k-1),\ 1,\ 1,\ (3k-1),\ 1,\ 1,\ (5k-1),\ 1,\ 1,\ (7k-1), \ldots]$$

for all $k \in \mathbf{N}$.

(a) Compute p_0, p_3, q_0, and q_3 for the above continued fraction. Your answers should be in terms of k.

(b) Condense three steps of the recurrence for the numerators and denominators of the above continued fraction. That is, produce a simple recurrence for r_{3n} in terms of r_{3n-3} and r_{3n-6} whose coefficients are polynomials in n and k.

(c) Define a sequence of real numbers by

$$T_n(k) = \frac{1}{k^n} \int_0^{1/k} \frac{(kt)^n (kt-1)^n}{n!}\ e^t dt.$$

i. Compute $T_0(k)$, and verify that it equals $q_0 e^{1/k} - p_0$.

ii. Compute $T_1(k)$, and verify that it equals $q_3 e^{1/k} - p_3$.

iii. Integrate $T_n(k)$ by parts twice in succession, as in Section 5.4, and verify that $T_n(k)$, $T_{n-1}(k)$, and $T_{n-2}(k)$ satisfy the recurrence produced in part 6b, for $n \geq 2$.

(d) Conclude that the continued fraction

$$[1,\ (k-1),\ 1,\ 1,\ (3k-1),\ 1,\ 1,\ (5k-1),\ 1,\ 1,\ (7k-1), \ldots]$$

represents $e^{1/k}$.

5.7 Let d be an integer that is coprime to 10. Prove that the decimal expansion of $\frac{1}{d}$ has a period equal to the order of 10 modulo d. (Hint: For every positive integer r, we have $\frac{1}{1-10^r} = \sum_{n\geq 1} 10^{-rn}$.)

5.8 Find a positive integer that has at least three different representations as the sum of two squares, disregarding signs and the order of the summands.

5.9 Show that if a natural number n is the sum of two rational squares it is also the sum of two integer squares.

5.10 (*) Let p be an odd prime. Show that $p \equiv 1, 3 \pmod 8$ if and only if p can be written as $p = x^2 + 2y^2$ for some choice of integers x and y.

5.11 Prove that of any four consecutive integers, at least one is not representable as a sum of two squares.

6
타원곡선 (Elliptic Curves)

타원곡선은 순수와 응용 정수론에서 중심적인 역할을 하는 정수론적인 대상이다. 정수론에서 합동수 문제-어떤 정수가 모든 변의 길이가 유리수인 직각삼각형의 면적인가?- 와 같은 심오한 문제가 자연스럽게 타원곡선에 관한 문제로 바뀐다. 유명한 Birch와 Swinnerton-Dyer 가설과 같은 다른 문제들은 타원곡선이 가질 것이라고 기대하는 신비한 구조를 묘사한다. 또 유한 아벨군을 타원곡선과 연결시킬 수가 있으며, 많은 경우에서 이 군들은 적절한 암호체계를 만들어준다. 특히 많은 사람들은 타원곡선이, 많은 응용에서 아주 유용한, 크기가 작은 키로 충분한 안전성을 보장해 줄 것으로 믿고 있다. 예를 들어 우편직인에 암호화 키를 인쇄하려고 한다면, 그 키의 길이가 짧으면 도움이 된다. 그뿐만 아니라 타원곡선을 이용하여 정수를 인수분해할 수 있는데, 앞에서 언급했듯이 이는 3.3절의 RSA의 공개키 암호를 교묘하게 공격하는데 아주 중요한 역할을 한다.

이 단원에서는 단원 1에서 단원 3에서 다룬 개념을 기반으로 하여 타원곡선을 간략하게 소개하고 여러가지 심오한 정리들과 개념들을 증명 없이 소개한다. 6.1절에서는, 타원곡선을 정의하고 타원곡선의 그림들을 그려본다. 그리고 나서 6.2절에서는 타원곡선들의 점들의 집합에 군의 구조를 넣는 방법을 설명한다. 6.3절과 6.4절은, 공개키 암호를 만들고 정수를 인수분해하는, 두 개의 암호 관련 문제에 타원곡선을 적용하는 방법에 관한 절이다. 마지막으로 절 6.5에서는 유리수 위에서의 타원곡선을 관찰하고, 1000년간 미해결 문제와 타원곡선과의 관계를 설명한다.

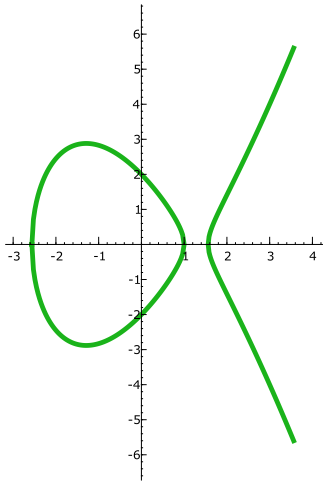


FIGURE 6.1. The elliptic curve $y^2 = x^3 - 5x + 4$ over $\mathbf{R}$

6.1 정의

정의 6.1.1 (타원곡선)**.** 체 K위의 **타원곡선(elliptic curve)**은 식

$$y^2 = x^3 + ax + b$$

으로 정의된 곡선이다. 단, $a, b \in K$이고 $-16(4a^3 + 27b^2) \neq 0$이다.

조건 $-16(4a^3 + 27b^2) \neq 0$은 타원곡선이 특이점(singular points)이 없을 조건이다. 이것은 우리들이 적용하고자 하는 문제에서는 아주 필수적인 조건이다. (Exercise 6.1 참조).

SAGE 예 6.1.2. `EllipticCurve` 명령어로 유리수체 $\mathbf{Q}$ 위에서 타원곡선을 생성하고 그림 6.1을 그린다.

```
sage: E = EllipticCurve([-5, 4])
sage: E
Elliptic Curve defined by y^2  = x^3 - 5*x + 4
over Rational Field
sage: P = E.plot(thickness=4,rgbcolor=(0.1,0.7,0.1))
sage: P.show(figsize=[4,6])
```

절 6.3에서는 정수를 인수분해하기위하여, 6.4절에서는 암호계를 만들기 위하여 유한체 위에서의 타원곡선을 이용한다. 다음 Sage 코드는 위수가 37인 유한체 위에서의 타원곡선을 생성해주고, 그림 6.2에서 예시한 것처럼 점들을 나타내준다.

```
sage: E = EllipticCurve(GF(37), [1,0])
sage: E
Elliptic Curve defined by y^2  = x^3 + x over
Finite Field of size 37
sage: E.plot(pointsize=45)
```

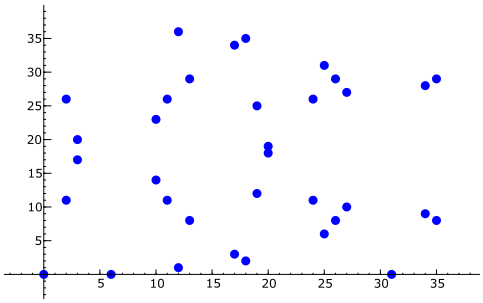


FIGURE 6.2. The elliptic curve $y^2 = x^3 + x$ over $\mathbf{Z}/37\mathbf{Z}$

6.2절에서는 집합 K위의 타원곡선 E의 점들에 항등원 역할을 하는 $\mathcal{O}$를 더한 집합

$$E(K) = \{(x, y) \in K \times K : y^2 = x^3 + ax + b\} \cup \{\mathcal{O}\}$$

위에 아벨군 구조를 자연스럽게 정의하려고 한다. 여기서 $\mathcal{O}$는 무한에 있는 점으로 생각하면 된다[1]. 그림 6.2는 유한체 $\mathbf{Z}/37\mathbf{Z}$ 위에서의 $y^2 = x^3 + x$을 만족하는 점들로 구성되어 있는데, 무한에 있는 점 $\mathcal{O}$는 구체적으로 그리지 않는다.

참조 6.1.3. 체 K의 표수가 2이면 (즉, K에서 $1+1=0$이면), 임의의 $a, b \in K$에 대하여 $-16(4a^3 + 27b^2) \in K$는 항상 0이다. 따라서, 정의 6.1.1에 의하면 K위에서의 타원곡선은 존재하지 않는다. 표수가 3인 경우도 유사한 문제가 발생한다. 대신, 우리가 다음과 같은 형태의 식

$$y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6,$$

으로 정의된 방정식을 더 일반적인 형태의 타원곡선들을 정의하면, 표수가 2와 3인 경우에도 바른 타원곡선을 얻을 수 있다. 이 곡선들이 암호론에서는 더 유명한 타원곡선들인데, 이유는 컴퓨터에 효율적으로 규현하는 것이 더 쉽기 때문이다.

---

[1] $\mathcal{O}$는 y축과 평행인 모든 직선들이 만나는 무한점으로 생각하면 된다

6.2 타원곡선의 군 구조

E는 $y^2 = x^3 + ax + b$으로 정의된 체 K위의 타원곡선이다. 먼저 $E(K)$ 위에 이항연산 $+$를 정의한다.

알고리듬 6.2.1 (타원곡선 군 연산)**.** $P_1, P_2 \in E(K)$일 때, 이 알고리듬은 세 번 째 점 $R = P_1 + P_2 \in E(K)$를 계산한다.

1. [Is $P_i = \mathcal{O}$?] 만약 $P_1 = \mathcal{O}$이면 $R = P_2$로 놓고, 만약 $P_2 = \mathcal{O}$이면 $R = P_1$으로 놓고 마친다. 그렇지 않으면 $(x_i, y_i) = P_i$로 쓴다.
2. [Negatives] 만약 $x_1 = x_2$이고 $y_1 = -y_2$이면, $R = \mathcal{O}$로 놓고 마친다.
3. [Compute λ] $\lambda = \begin{cases} (3x_1^2 + a)/(2y_1) & \text{if } P_1 = P_2, \\ (y_1 - y_2)/(x_1 - x_2) & \text{otherwise.} \end{cases}$
   라 놓는다.
4. [Compute Sum] 그러면 $R = \big(\lambda^2 - x_1 - x_2, -\lambda x_3 - \nu\big)$인데, 이 때 $x_3 = \lambda^2 - x_1 - x_2$는 R 의 x-성분이고, $\nu = y_1 - \lambda x_1$이다.

단계 3에서 $P_1 = P_2$이면 $y_1 \neq 0$이다. 그렇지 않으면 그 전 단계에서 끝났엇을 것이기 때문이다.

정리 6.2.2. *알고리듬 6.2.1에서 정의된 연산 $+$는 집합 $E(K)$에 항등원이 $\mathcal{O}$인 아벨군 구조를 준다.*

왜 이 정리가 참인지를 논하기 전에 $+$를 기하학적으로 재해석하여 쉽게 시각화하여 보자. $P_1 + P_2$는 P_1과 P_2를 지나는 직선 L이 E와 만나는 점 R의 x-축에 대칭인 점이다. (이 설명은 경우 1과 2, 또 $P_1 = P_2$일 때도 적절하게 적용될 수 있다.) 그림 6.3은 타원 곡선 $y^2 = x^3 - 5x + 4$에서 덧셈 $(0, 2) + (1, 0) = (3, 4)$을 앞에서 설명한 대로 그림으로 보여주고 있다.

SAGE 예 6.2.3. 타원곡선 $y^2 = x^3 - 5x + 4$을 Sage에서 생성하고 두 점 $P = (1, 0)$와 $Q = (0, 2)$를 더한다. 또 $P+P$를 계산하는데, 물론 이 값은 무한대의 점 $\mathcal{O}$인데, Sage에서는 $(0 : 1 : 0)$로 나타낸다. 그리고 $P+Q+Q+Q+Q$를 계산하는데, 그 값은 놀라울 정도로 크다.

```
sage: E = EllipticCurve([-5,4])
sage: P = E([1,0]); Q = E([0,2])
sage: P + Q
(3 : 4 : 1)
sage: P + P
(0 : 1 : 0)
sage: P + Q + Q + Q + Q
(350497/351649 : 16920528/208527857 : 1)
```

군연산의 기하적 설명을 더 분명하게 하기 위하여 다음 기초정리를 소개한다.

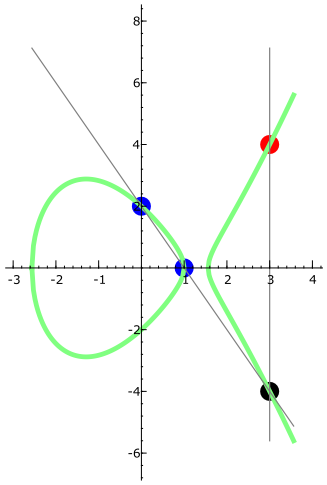


FIGURE 6.3. The Group Law: $(1,0)+(0,2)=(3,4)$ on $y^2=x^3-5x+4$

기초정리 6.2.4 (기하적 군연산 정의)**.** *$P_i=(x_i,y_i)$, $i=1,2$는 타원곡선 $y^2=x^3+ax+b$위의 두 점이고 $x_1\neq x_2$라고 하자. L은 P_1과 P_2를 지나는 유일한 직선이다. 그러면 L은 E와 P_1과 P_2외에 정확히 한 점*

$$Q=\left(\lambda^2-x_1-x_2,\quad \lambda x_3+\nu\right),$$

에서 더 만난다. 단, $\lambda=(y_1-y_2)/(x_1-x_2)$, $\nu=y_1-\lambda x_1$.

증명 P_1과 P_2를 지나는 직선 L의 식은 $y=y_1+(x-x_1)\lambda$이다. 이 식을 $y^2=x^3+ax+b$에 대입하면

$$(y_1+(x-x_1)\lambda)^2=x^3+ax+b.$$

간단히 하면 $f(x)=x^3-\lambda^2x^2+\cdots=0$꼴로 쓸 수 있는데, x항과 상수항은 꼭 필요하지 않으므로 생략하였다. P_1과 P_2는 $L\cap E$의 점이므로, x_1과 x_2는 다항식 f의 해이다. 따라서 f는 $(x-x_1)(x-x_2)$로 나누어 떨어지므로, $f=\prod_{i=1}^{3}(x-x_i)$로 쓸 수 있고, 전개하면 $x_1+x_2+x_3=\lambda^2$가 된다. 따라서, 우리가 주장한대로 $x_3=\lambda^2-x_1-x_2$을 얻는다. 또 L의 식으로부터 $y_3=y_1+(x_3-x_1)\lambda=\lambda x_3+\nu$을 얻는다. □

정리 6.2.2를 증명한다는 것은 $+$가 아벨군의 네 개의 공리인 $\mathcal{O}$가 항등원임을 확인하고, 역원의 존재성과 교환과 결합법칙이 성립함을 보이는 것이다. 정의로부터 $(x,y)+(x,-y)=\mathcal{O}$이므로 역원은 항상 존재한다. 두 점 P_1과 P_2의 순서를 바꾸더라도 이 두 점을 지나는 직선은 동일하므로 교환법칙 역시 성립한다.

증명하기 어려운 성질은 결합법칙이다. 즉, 세 점 P_1,P_2,P_3 에 대하여

$$(P_1+P_2)+P_3=P_1+(P_2+P_3)$$

을 보이는 것이다. 첫 번째 방법은 +을 모든 경우를 포함하여 기하적인 방법으로 다시 설명한 후, 결합의 성질을 평면(사영)기하의 문제로 바꾸어 증명한다. 이 접근은 아주 고전적인 접근 방법인데, [51]에 적절한 수준으로 아주 아름답게 자세히 설명되어있다. 또 다른 방법은 +의 공식을 이용하여, 손으로 계산하는 것은 아주 지루한 일이긴 하지만, 양쪽의 연산 값이 같음을 확인하는 것이다. 물론 컴퓨터를 이용하려 계산할 수는 있지만 이것 역시 지루할 수 있다. 세 번째 접근 방법은 ([48]이나 [21]참조) 대수곡선의 모든 점들의 집합으로 생성되는 자유아벨군(free abelian group)인 divisor군에 대한 일반적인 이론을 이용하는 건데, 여기에서는 연산의 결합법칙은 아주 자연스러운 따름정리일 뿐이다. 이런 점에서 세 번 째 접근이 최선이긴 하지만 이를 위하여는 수학의 새로운 세상을 많이 소개해야 하고 이 책의 의도를 많이 벗어나므로 생략한다.

SAGE 예 6.2.5. 다음 Sage 예에서는 알고리듬 6.2.1을 이용하여, 유리수체 $\mathbf{Q}$위의 타원곡선상의 임의의 세 점 P_1, P_2, P_3가, $P_1, P_2, P_3, P_1 + P_2, P_2 + P_3$가 다 다르고 항등원이 아니라면, 결합법칙을 만족함을 보인다. 변수가 8 개인 다항식환 R을 먼저 정의함으로써 이작업을 수행한다.

```
sage: R.<x1,y1,x2,y2,x3,y3,a,b> = QQ[]
```

x_i가 만족해야 할 조건를 정의하고 그 조건들을 만족하는 잉여환(factor ring) Q 를 정의한다. 잉여환은 정수환 $\mathbf{Z}$로 부터 $\mathbf{Z}/n\mathbf{Z}$를 생성하는 것과 유사하다. $\mathbf{Z}/n\mathbf{Z}$에서는 $\mathbf{Z}$에서와 궁극적으로 같은 연산이지만 $n\mathbf{Z}$의 원소들을 0으로 본다는 점이 다른 점이었다.)

```
sage: rels = [y1^2 - (x1^3 + a*x1 + b),
...           y2^2 - (x2^3 + a*x2 + b),
...           y3^2 - (x3^3 + a*x3 + b)]
...
sage: Q = R.quotient(rels)
```

모든 점이 다르다고 가정하고 군 연산을 정의한다.

```
sage: def op(P1,P2):
...       x1,y1 = P1;  x2,y2 = P2
...       lam = (y1 - y2)/(x1 - x2); nu  = y1 - lam*x1
...       x3 = lam^2 - x1 - x2; y3 = -lam*x3 - nu
...       return (x3, y3)
```

세 점을 정의하고, $P_1 + (P_2 + P_3)$와 $(P_1 + (P_2 + P_3))$를 각각 계산하고, 두 결과의 x-성분과 y-성분이 같음을 확인한다.

```
sage: P1 = (x1,y1); P2 = (x2,y2); P3 = (x3,y3)
sage: Z = op(P1, op(P2,P3)); W = op(op(P1,P2),P3)
sage: (Q(Z[0].numerator()*W[0].denominator() -
...          Z[0].denominator()*W[0].numerator())) == 0
True
```

```
sage: (Q(Z[1].numerator()*W[1].denominator() -
...           Z[1].denominator()*W[1].numerator())) == 0
True
```

6.3 타원곡선을 이용한 정수의 인수분해

1987년 Hendrik Lenstra가 정수를 인수분해하는 강력한 알고리듬인 타원곡선방법 ECM을 소개하는 획기적인 논문[32]을 발표하였다. Lenstra의 방법은 [51, §IV.4], [15, §VIII.5], [10, §10.3]에도 설명되어있다.

H. Lenstra

Lenstra의 알고리듬은 정수 N의 "중간크기의 (medium-sized)"의 인수를 찾는데 최적화 되어 있다. 오늘날 중간크기란 10에서 40 자릿수 사이의 수를 의미한다. ECM 방법은 RSA 도전수를 인수분해하는데에 *바로* 이용되지는 않지만 (1.1.3절 참조), 어떤 수의 약수를 찾아내는데 최선의 알고리듬으로 알려진 "number field sieve"에서 결정적인 역할을 하는 보조적인 수의 인수를 찾는데에 사용된다. 또, ECM을 구현하는데는 메모리도 적게 필요하다.

6.3.1 Pollard의 $(p-1)$-방법

Lenstra의 발견은 이 절에서 소개하려고 하는 Pollard의 $(p-1)$-방법에 영향을 받아 만들어졌다.

정의 6.3.1 (Power Smooth)**.** B는 양의 정수이다. 양의 정수 n이 $n = \prod p_i^{e_i}$로 인수분해될 때, 모든 i에 대하여 $p_i^{e_i} \leq B$를 만족하면 n는 **B-power smooth** 라고 정의한다.

예를 들어, $30 = 2 \cdot 3 \cdot 5$는 $B = 5, 7$로 두면 B-power smooth이지만, $150 = 2 \cdot 3 \cdot 5^2$는 5-power smooth는 아니다 ($B = 25$-power smooth는 됨).

Pollard의 $p-1$ 방법과 타원곡선방법에 다음 알고리듬을 이용하려고 한다.

알고리듬 6.3.2 (Least Common Multiple of First B Integers)**.** 다음은 양의 정수 B를 선택하여 B가지의 모든 양의 정수들의 최소공배수흘 구하는 알고리듬이다.

1. [Sieve] 예를 들어 소수만을 걸러내는 체의 알고리듬(알고리듬 1.2.3)을 이용하여 $p \leq B$인 모든 소수들의 목록 P을 만든다.
2. [Multiply] $\prod_{p \in P} p^{\lfloor \log_p(B) \rfloor}$를 계산하고 출력한다.

증명 $m = \mathrm{lcm}(1, 2, \ldots, B)$이라 놓자. 그러면,

$$\mathrm{ord}_p(m) = \max(\{\mathrm{ord}_p(n) : 1 \leq n \leq B\}) = \mathrm{ord}_p(p^r)$$

이라 놓으면 p^r은 $p^r \le B$을 만족하는 가장 큰 p의 거듭제곱이다. $p^r \le B < p^{r+1}$이므로, $r = \lfloor \log_p(B) \rfloor$이다. □

SAGE 예 6.3.3. 알고리듬 6.3.2를 Sage에서 구현하고 $B = 100$일 때 위의 알고리듬과 단순한 알고리듬을 이용하여 최소공배수를 구한다. 아래에서 $\log_p(B)$를 빠르게 계산하기 위하여 `math.log`를 사용한다.

```
sage: def lcm_upto(B):
...       return prod([p^int(math.log(B)/math.log(p))
...                    for p in prime_range(B+1)])
sage: lcm_upto(10^2)
69720375229712477164533808935312303556800
sage: LCM([1..10^2])
69720375229712477164533808935312303556800
```

위에 구현한 알고리듬 6.3.2은 Sage에서 $B = 10^6$일 때 약 1초가 걸린다.

이제 우리가 인수분해하고 싶은 수가 N이라고 하자. N의 의미있는 약수를 찾기 위하여 다음과 같이 Pollard의 $(p-1)$-방법을 사용한다. 먼저, 6 자릿수를 넘지않는 적당한 양의 정수 B를 선택한다. 그리고 N의 소수 약수 p중에 $p-1$이 B-power smooth가 되는 p가 존재한다고 가정한다. 그러면 만약 1보다 큰 수 a가 p로 나누어지지 않으면, 정리 2.1.20에 의하여

$$a^{p-1} \equiv 1 \pmod{p}$$

가 성립해야 한다. $m = \mathrm{lcm}(1, 2, 3, \ldots, B)$으로 놓고, $p-1$이 B-power smooth라는 가정은 $p - 1 \mid m$을 유도하므로

$$a^m \equiv 1 \pmod{p}$$

이 성립한다. 그러므로

$$p \mid \gcd(a^m - 1, N) > 1$$

이 성립한다. 만약 $\gcd(a^m - 1, N) < N$이면, $\gcd(a^m - 1, N)$도 N의 비자명 약수이다. 만약 $\gcd(a^m - 1, N) = N$이면, 소수의 멱으로 표현되는 N의 모든 약수 q^r이 $a^m \equiv 1 \pmod{q^r}$를 만족한다. 이 경우에는 위의 단계를, 더 작은 B를 선택하고, 가능하면 a도 다른 값을 선택하여, 반복한다. 또 처음 시작할 때 N이 다른 수의 거듭제곱 M^r 형태인지를 확인하고, 그런 경우에는 처음부터 N을 M으로 바꾸어 시작한다. 이 알고리듬을 만들면 다음과 같다.

알고리듬 6.3.4 (Pollard의 $p-1$ 방법)**.** 양의 정수 N과 상한 B를 입력하였을 때, 이 알고리듬은 N의 비자명 약수 g를 찾는 시도를 한다. ($p \mid g$인 각각의 소수 p는 $p - 1$이 B-power smooth인 성질을 가질 가능성이 있다.)

1. [Compute lcm] 알고리듬 6.3.2을 사용하여 $m = \mathrm{lcm}(1, 2, \ldots, B)$을 계산한다.
2. [Initialize] $a = 2$로 놓느다.

3. [Power and gcd] $x = a^m - 1 \pmod N$와 $g = \gcd(x, N)$를 계산한다.
4. [Finished?] 만약 g가 1도 N도 아니면, g를 출력하고 마친다.
5. [Try Again?] 만약 (예를 들어) $a < 10$이라면, a를 $a+1$로 바꾸고, 단계 3으로 간다. 그렇지 않으면 마친다.

B를 고정한 후, 알고리듬 6.3.4은, N이 $p-1$이 B-power smooth인 소수 p로 나누어지면, N을 종종 인수분해한다. 대략적으로 10^{15}부터 $10^{15}+10000$사이의 구간에서 약 15%의 소수 p만이 $p-1$이 10^6-power smooth이다. 따라서 이 범위의 수의 85%는 $B = 10^6$일 때 Pollard 방법으로 15-자릿수의 소수약수를 찾을 수 없다. (Exercise 6.10 참조). 여기서는 타원곡선을 이용한 인수분해를 설명하고자 하므로 Pollard 방법을 더 이상 분석하지는 않는다.

다음 예들은 Pollard $(p-1)$-방법을 설명한다.

예 6.3.5. 이 예에서 Pollard 방법은 완벽하게 작동한다. $N = 5917$이라 놓자. $B = 5$로 잡은 Pollard의 $p-1$ 방법으로 N을 인수분해한다. 먼저 $m = \mathrm{lcm}(1,2,3,4,5) = 60$를 계산하고, $a = 2$로 잡아

$$2^{60} - 1 \equiv 3416 \pmod{5917}$$

와

$$\gcd(2^{60} - 1, 5917) = \gcd(3416, 5917) = 61$$

를 계산하면, 5917의 약수 61을 구할 수 있다.

예 6.3.6. 이 예에서는 B를 더 큰 정수로 교환한다. $N = 779167$의 인수를 구하기 위하여, $B = 5$, $a = 2$로 잡자, 그러면 $m = 6$이다.

$$2^{60} - 1 \equiv 710980 \pmod{779167}$$

이므로 $\gcd(2^{60} - 1, 779167) = 1$이다. $B = 15$로 잡으면

$$m = \mathrm{lcm}(1, 2, \ldots, 15) = 360360$$

가 되고,

$$2^{360360} - 1 \equiv 584876 \pmod{779167},$$

그리고

$$\gcd(2^{360360} - 1, N) = 2003$$

를 계산함으로써 779167의 약수 2003을 찾는다.

예 6.3.7. 이 예에서는 더 작은 정수 B를 사용한다. $N = 4331$이고 $B = 7$이라면, $m = \mathrm{lcm}(1, 2, \ldots, 7) = 420$이다. 또

$$2^{420} - 1 \equiv 0 \pmod{4331},$$

$\gcd(2^{420} - 1, 4331) = 4331$이므로 4331의 인수는 얻지 못한다. 그런데 B를 5로 바꾸면, Pollard의 방법이 효력을 나타낸다. 즉,

$$2^{60} - 1 \equiv 1464 \pmod{4331},$$

$\gcd(2^{60} - 1, 4331) = 61$이므로 4331을 이제 인수분해할 수 있다.

예 6.3.8. 이 예에서는 $a = 2$로는 원하는 결과를 얻을 수 없지만, $a = 3$으로는 가능하다. $N = 187$일 때, $B = 15$로 가정하고, $m = \mathrm{lcm}(1, 2, \ldots, 15) = 360360$을 계산한다. $a = 2$로 실행하면,

$$2^{360360} - 1 \equiv 0 \pmod{187},$$

$\gcd(2^{360360} - 1, 187) = 187$이므로 187의 약수를 얻지는 못한다. 이제 $a = 3$으로 Pollard의 방법을 실행하면,

$$3^{360360} - 1 \equiv 66 \pmod{187},$$

$\gcd(3^{360360} - 1, 187) = 11$이다. 따라서 187은 11로 나누어진다. 즉, $187 = 11 \cdot 17$이다.

SAGE 예 6.3.9. Sage에서 Pollard의 $(p-1)$-방법의 구현하고 이를 이용하여 위의 모든 계산을 수행한다.

```
sage: def pollard(N, B=10^5, stop=10):
...       m = prod([p^int(math.log(B)/math.log(p))
...                 for p in prime_range(B+1)])
...       for a in [2..stop]:
...           x = (Mod(a,N)^m - 1).lift()
...           if x == 0: continue
...           g = gcd(x, N)
...           if g != 1 or g != N: return g
...       return 1
sage: pollard(5917,5)
61
sage: pollard(779167,5)
1
sage: pollard(779167,15)
2003
sage: pollard(4331,7)
1
sage: pollard(4331,5)
61
sage: pollard(187, 15, 2)
1
sage: pollard(187, 15)
11
```

6.3.2 타원곡선방법의 동기

양의 정수 B를 고정한다. $N = pq$이고 p와 q는 소수이고, $p-1$과 $q-1$은 둘 다 B-power smooth가 아니라고 가정한다. 그러면 Pollard의 $(p-1)$-방법은 작동하지 않을 가능성이 크다. 예를 들어, $B = 20$으로 잡고 $N = 59 \cdot 101 = 5959$

이라고 가정한다. $59-1=2\cdot 29$도 $101-1=4\cdot 25$도 B-power smooth는 아님을 확인하자. $m=\mathrm{lcm}(1,2,3,\ldots,20)=232792560$을 계산한 후

$$2^m-1\equiv 5944 \pmod{N}$$

와 $\gcd(2^m-1,N)=1$를 계산하였지만 N의 약수를 찾지는 못한다.

위에서 지적하였듯이, 문제는 $p=59$이든 $p=101$이든, $p-1$이 20-power smooth가 아니기 때문에 발생한다. 그러나 $p-2=3\cdot 19$는 20-power smooth가 된다! Lenstra의 ECM은 위수가 $p-1$인 $(\mathbf{Z}/p\mathbf{Z})^*$을 $\mathbf{Z}/p\mathbf{Z}$ 위에서의 타원곡선이 대신한다. 타원곡선의 크기는, 적당한 음수가 아닌 정수 $s<2\sqrt{p}$가 있어,

$$\#E(\mathbf{Z}/p\mathbf{Z})=p+1\pm s$$

임이 정리로 알려져 있다 ([48, §V.1]. 또, s가 취할 수 있는 모든 값을 갖는 타원곡선이 실제 존재하는 것도 알려져 있는데, "complex multiplication theory"을 사용하여 확인할 수 있다. 예를 들어, 만약 E가 $\mathbf{Z}/59\mathbf{Z}$위에서

$$y^2=x^3+x+54$$

로 정의된 타원곡선이라면, 점들을 나열해 봄으로써 $E(\mathbf{Z}/59\mathbf{Z})$는 위수가 57인 순환군임을 보일 수 있다. $s\le 15$를 만족할 때 $59+1\pm s$로 주어지는 정수들의 집합에는 5-power smooth인 수들이 14개가 들어있으므로, 타원곡선으로 작업 하는 것이 훨씬 더 유연함을 알 수 있다. 예를 들어 $60=59+1+0$은 5-power smooth이고 $70=59+1+10$은 7-power smooth이다.

6.3.3 Lenstra의 타원곡선 인수분해 방법

알고리듬 6.3.10 (타원곡선 인수분해 방법)**.** 인수분해할 양의 정수 N과 제어할 정수 B를 주면, 이 알고리듬은 N의 비자명 인수 g를 찾는 시도를 하거나 실패하면 "Fail"을 출력한다.

1. [Compute lcm] 알고리듬 6.3.2를 사용하여 $m=\mathrm{lcm}(1,2,\ldots,B)$을 계산한다.
2. [Choose Random Elliptic Curve] $4a^3+27\in(\mathbf{Z}/N\mathbf{Z})^*$을 만족하는 적당한 수 $a\in\mathbf{Z}/N\mathbf{Z}$를 선택한다. 그러면 $P=(0,1)$은 타원곡선 $\mathbf{Z}/N\mathbf{Z}$ 위의 $y^2=x^3+ax+1$의 한 점이다.
3. [Compute Multiple] 알고리듬 2.3.13을 타원곡선의 연산에 적용하여 mP를 계산하려고 시도한다. 만약 어떤 점에서 그 점의 합을 계산할 수 없으면, 알고리듬 6.2.1의 Step 3에 나타나는 어떤 점의 분모가 N과 서로 소가 아니라는 것을 의미하는데, 이는 이 분모와 N의 최대공약수가 1이상이라는 것을 의미하므로 최대공약수를 계산한다. 만약 g가 비자명 약수이면 g를 출력한다. 만약 모든 분모가 N과 서로 소이면 "Fail"을 출력한다.

알고리듬 6.3.10이 선택한 한 타원곡선에서 실패하면 Pollard의 $(p-1)$-방법에서는 이용할 수 없었던 다른 타원곡선을 선택하여 같은 작업을 반복할 수

있다. Pollard의 $(p-1)$-방법에서는 항상 같은 군 $(\mathbf{Z}/N\mathbf{Z})^*$에서 작업을 해야 하는데, ECM에서는 많은 타원곡선 E의 군인 $E(\mathbf{Z}/N\mathbf{Z})$를 시도할 수 있다. 위에서 언급했듯이 $\mathbf{Z}/p\mathbf{Z}$위의 $E(\mathbf{Z}/N\mathbf{Z})$의 원소의 개수는 $p+1-t$이고 이 때 $|t| < 2\sqrt{p}$이고 실제 그 범위의 모든 t가 실제 일어나므로, $p+1-t$가 B-power smooth가 되면 알고리듬 6.3.10는 제대로 작동될 기회를 얻는다.

6.3.4 예제

타원곡선의 식을 간단하게

$$y^2 = x^3 + ax + 1$$

로 놓으면 $P = (0,1)$은 항상 타원곡선의 점이다.

타원곡선 방법을 사용하여 $N = 5959$ 을 인수분해한다. m을 계산하여 2진법으로 쓰면

$$m = \mathrm{lcm}(1,2,\ldots,20) = 232792560 = 11011110000000100000111110000_2$$

이다. 이 때 x_2는 x를 이진법으로 쓴 것이다. 먼저, $a = 1201$을 임의로 선택하고, $\mathbf{Z}/5959\mathbf{Z}$ 위의 $y^2 = x^3+1201x+1$를 고려한다. 알고리듬 6.2.1에 있는 $P+P$를 구하는 공식을 이용하여 $i \in B = \{4,5,6,7,8,13,21,22,23,24,26,27\}$에 대하여 $2^i \cdot P = 2^i \cdot (0,1)$ 를 계산한다. 결과적으로 이 계산의 어떤 단계에서도 5959와 서로 소가 아닌 수가 분모에 나타나지 않았기 때문에 N을 인수분해 하지 못한다. 다음 $a = 389$를 시도하는데 계산이 어떤 단계에서 $P = (2051,5273)$와 $Q = (637,1292)$를 더하게 된다. 연산 과정 중에 $(\mathbf{Z}/5959\mathbf{Z})^*$에서 $\lambda = (y_1-y_2)/(x_1-x_2)$ 를 계산해야하는데, $x_1-x_2 = 1414$이고 $\gcd(1414,5959) = 101$이므로 더 이상 계산을 할 수가 없다. 따라서 덕분에 우리는 5959의 약수 101을 찾아낸다.

SAGE 예 6.3.11. Sage에 타원곡선을 이용한 인수분해 알고리듬을 구현하고 이를 이용하여 위에서 설명한 예와 다른 예들을 실행한다.

```
sage: def ecm(N, B=10^3, trials=10):
...       m = prod([p^int(math.log(B)/math.log(p))
...                 for p in prime_range(B+1)])
...       R = Integers(N)
...       # Make Sage think that R is a field:
...       R.is_field = lambda : True
...       for _ in range(trials):
...           while True:
...               a = R.random_element()
...               if gcd(4*a.lift()^3 + 27, N) == 1: break
...           try:
...               m * EllipticCurve([a, 1])([0,1])
...           except ZeroDivisionError, msg:
...               # msg: "Inverse of <int> does not exist"
```

```
...                 return gcd(Integer(str(msg).split()[2]), N)
...         return 1
sage: set_random_seed(2)
sage: ecm(5959, B=20)
101
sage: ecm(next_prime(10^20)*next_prime(10^7), B=10^3)
10000019
```

6.3.5 발견적 설명

N은 양의 정수이고, 상황을 간단하게 하기 위하여, N은 서로 다른 소수 p_i들의 곱인 $N = p_1 \cdots p_r$이라 가정한다. 보조정리 2.2.5로부터 자연스러운 동형사상

$$f : (\mathbf{Z}/N\mathbf{Z})^* \longrightarrow (\mathbf{Z}/p_1\mathbf{Z})^* \times \cdots \times (\mathbf{Z}/p_r\mathbf{Z})^*$$

이 존재한다. Pollard의 방법을 사용할 때, $a \in (\mathbf{Z}/N\mathbf{Z})^*$를 선택하고, a^m을 계산하고, 그런 후 $\gcd(a^m - 1, N)$을 계산한다. 이 gcd는 정확히 $a^m \equiv 1 \pmod{p_i}$을 만족하는 p_i에 의해 나누어진다. Pollard의 방법을 위의 동형사상을 이용하여 재해석하기 위하여, $f(a) = (a_1, \ldots, a_r)$라고 하자. 그러면 $(a_1^m, \ldots, a_r^m) = f(a^m)$이고, $\gcd(a^m - 1, N)$을 나누는 p_i는 $a_i^m \equiv 1 \pmod{p_i}$를 만족한다. 정리 2.1.20로부터 이 p_i들은, $m = \mathrm{lcm}(1, \ldots, B)$일 때, $p_j - 1$이 B-power smooth인 모든 소수 p_j를 포함한다.

우리의 알고리듬에서는 N이 소수라고 가정하고 타원곡선을 정의한 후 모순을 찾아내는 것이므로, 우리는 N이 합성수일 때는 $E(\mathbf{Z}/N\mathbf{Z})$를 정의하지 않는다. 그러나 이 문단의 나머지 부분에서는, 마치 $E(\mathbf{Z}/N\mathbf{Z})$이 의미가 있는 것처럼 여기고, 경험으로 알게 된 Lenstra와 Pollard의 방법 사이의 관계를 설명한다. 의미있는 중요한 차이는 동형사상 f를 동형사상

$$\text{“}g : E(\mathbf{Z}/N\mathbf{Z}) \to E(\mathbf{Z}/p_1\mathbf{Z}) \times \cdots \times E(\mathbf{Z}/p_r\mathbf{Z})\text{”}$$

가 대신하는데, 이 때 E는 $y^2 = x^3 + ax + 1$으로 정의되고, Pollard의 방법에서의 a는 $P = (0, 1)$가 대신하는 것이다. 우리가 $E(\mathbf{Z}/N\mathbf{Z})$를 정의하지 않았다는 것을 강조하기 위하여 동형사상 g에 따옴표를 사용하였다. 타원곡선 인수분해 알고리듬을 구현할 때, mP를 계산하려고 시도하는데, 만약 우리의 계산 중에 얻은 어떤 점 Q에 대하여 $f(Q)$의 성분 중 하나가 $\mathcal{O}$이 되면, 우리는 N의 약수를 발견하게 된다.

6.4 타원곡선 암호

타원곡선을 암호론에 이용하려는 아이디어는 1980년대 중반에 Neil Koblitz와 Victor Miller에 의해서 독립적으로 제안되었다. 이 절에서는 $(\mathbf{Z}/p\mathbf{Z})^*$대신에 타원곡선을 사용하는 Diffie-Hellman 공개키와 유사한 암호를 소개한다. 그런 후 ElGamal 타원곡선 암호를 논한다.

6.4.1 Diffie-Hellman의 타원곡선 버전

Diffie-Hellman 의 키 교환 프로토콜은 큰 수정없이 타원곡선에 적용가능하다. Michael과 Nikita는 다음과 같이 비밀키에 동의한다.

1. Michael과 Nikita는 소수 p, $\mathbf{Z}/p\mathbf{Z}$위의 타원곡선과 곡선위의 점 $P \in E(\mathbf{Z}/p\mathbf{Z})$를 함께 선택한다.
2. Michael는 m을 몰래 선택하고 mP를 전송한다.
3. Nikita는 n을 몰래 선택하여 nP를 보낸다.
4. 비밀키는 nmP인데, 이 값은 Michael과 Nikita만이 계산할 수 있다.

이산로그문제를 풀지 못하면 짐작컨데 적은 nmP를 계산할 수 없다 (문제 3.2.2와 다음 절 6.4.3 참조).

E, P,그리고 p를 잘 선택하면, $E(\mathbf{Z}/p\mathbf{Z})$에서의 이산로그문제는 $(\mathbf{Z}/p\mathbf{Z})^*$에서의 이산로그문제보다 훨씬 더 어렵다는 것이 경험적으로 알려져 있다. (타원곡선 이산로그문제에 대해 더 알고 싶으면 절 6.4.3를 참조)

6.4.2 ElGamel 암호와 디지털 권한 관리

이 절에서는 "Digital Rights Management" (DRM) system을 깬 컴퓨터 해커 Beale Screamer의 논문을 참조하여 타원곡선위에서 잘 작동하는 ElGamal 암호에 관해 설명한다.

DRM에서 사용한 소수와 타원곡선의 식은 다음과 같다.

$$p = 7859631023794288223766947894468973962074985689 51$$

$$y^2 = x^3 + 3176890812513255034763174764138276932727469559 27x + 7905289660787875871812057202571853543210065193 4.$$

소수 p의 16진법으로 표현하면

$$89ABCDEF012345672718281831415926141424F7.$$

그러면

$$\#E(k) = 78596310237942882237669302488171495761268615742 9$$

이고, 군 $E(k)$는 생성원이

$$B = (7715072162626498261706482685655798899077692541 76, 3901575102465566285252794592665149955625331966 55)$$

인 순환군이다.

우리의 영웅인 Nikita와 Michael은 그들이 싸우는 테러리스트로 나가 있지 않을 때에는 디지털 음악을 함께 감상한다. Nikita가 그녀의 컴퓨터에 DRM 소프트웨어를 인스톨할 때, 개인키

$$n = 6708050311399105135175272076930604563002170544 73,$$

이 생성되어 비트와 파일 조각에 숨어있다. Nikita가 Juno Reactor의 히트곡 `juno.wma`를 연주하기 위하여, Nikita는 인터넷으로 그 음악을 파는 웹싸이트에 접속한 후 신용카드의 번호를 보내면 그 웹싸이트는 Nikita 가 사용권 파일을 다운로드 할 수 있도록 하여 오디오가 `juno.wma`의 장금을 해재하여 음악을 연주하게 한다.

아래에서 알게 되겠지만 사용권 파일은 군 $E(k)$에서 ElGamal 공개키 암호을 이용하여 생성된다. Nikita는 이제 그녀의 사용권 파일을 `juno.wma`의 장금을 해제하기 위하여 사용할 수 있다. 그러나 그녀가 `juno.wma`와 사용권 파일을 Michael과 공유할 때, 실망스럽게도 Michael의 컴퓨터에서는 Nikita의 사용권 파일이 작동하지 않아 `juno.wma`를 들을 수가 없다. 이유는 Michael의 컴퓨터는 Nikita의 컴퓨터의 개인키(위에서 설명한 정수 n)를 알지 못하기 때문에 Michael의 컴퓨터는 사용권 파일을 해독할 수 없다.

이제 ElGamal 암호체계를 자세히 설명하려고 하는데, 이 ElGamal 암호는 군 $E(\mathbf{Z}/p\mathbf{Z})$에서 잘 구현된다. ElGamal 암호를 예로 설명하기 위하여 Nikita가, 누구든지 그녀에게 보낼 메세지를 암호화할 수 있도록, ElGamal 암호체계를 어떻게 구성했는지 예를 들어 설명한다. Nikita는 소수 p, $\mathbf{Z}/p\mathbf{Z}$위의 타원곡선 E, 그리고 점 $B \in E(\mathbf{Z}/p\mathbf{Z})$를 선택한 후, p, E,그리고 B를 공개한다. 또 그녀의 비밀키 n을 무작위로 선택하여 nB를 공개한다. 그러면 그녀의 공개키는 (p, E, B, nB)이다.

Michael이 Nikita에게 보낼 메세지를 암호화하고 싶다고 가정한다. 만약 그 메세지가 타원곡선상의 점 $P \in E(\mathbf{Z}/p\mathbf{Z})$로 코딩이 되었다면 Michael도 정수 r을 무작위로 선택하여 $E(\mathbf{Z}/p\mathbf{Z})$ rB와 $P + r(nB)$를 계산한다. 두 점 $(rB, P + r(nB))$가 P의 암호문이다. 이 암호문을 해독하기 위하여 Nikita는 그녀의 비밀키 n을 가지고 rB를 n배하여 $n(rB) = r(nB)$를 계산한 후 $P + r(nB)$에서 이 값을 빼서

$$P = P + r(nB) - r(nB)$$

를 구한다.

참조 6.4.1. 군 $(\mathbf{Z}/p\mathbf{Z})^*$에서도 마찬가지로 ElGamal 암호체계를 만들 수 있다. $(\mathbf{Z}/p\mathbf{Z})^*$.

우리 이야기로 다시 돌아오면, Nikita의 사용권 파일은 암호화된 파일이다. 이 파일은 두 점 $(rB, P + r(nB))$을 포함하는데, rB와 $P + r(nB)$는 다음과 같다.

$$rB = (17967100321831574638502665573308604498219442466 0,$$
$$69783438535968636824930128267514183093517631471 8);$$

$$P + r(nB) = (137851038548264467372645158093004000343639118915,$$
$$1108485892286762240572292302235808150242248756 99).$$

Nikita의 컴퓨터가 `juno.wma`를 재생할 때, 비밀키

$$n = 67080503113991051351752720769306045630021705 4473$$

이 메모리에 같이 다운로드되고

$$n(rB) = (32890139351873263757711565060176868104404071 5701,$$
$$5869478380878159936013505654887888462038879881 62).$$

를 계산한다. 이 값을 $P + r(nB)$에서 빼서

$$P = (14489646124220757767,$$
$$6693377803732840962748951366181946044696968300 74).$$

를 알아낸다. x-성분인 14489646124220757767가 `juno.wma`의 잠금을 해제하는 키이다.

만약 Nikita가 그녀의 컴퓨터에서 생성되는 개인키 n을 안다면 P를 계산할 수 있고, `juno.wma`의 잠금을 해제하여 Michael과 함께 음악을 공유할 수 있다. Beale Screamer는 이 암호체계를 구현할 때 Nikita가 n을 알아낼 수도 있는 약점을 발견하였는데, 이 사실은 니키타의 컴퓨터에 n이 저장되어 있으므로 너무 놀랄만한 일은 아니다.

SAGE 예 6.4.2. Sage에서 다음 예를 한다.

```
sage: p = 78596310237942882237669478944689739620749856895 1
sage: E = EllipticCurve(GF(p), \
...     [31768908125132550347631747641382769327274695592 7,
...      79052896607878758718120572025718535432100651934])
sage: E.cardinality()
78596310237942882237669302488171495761268615742 9
sage: E.cardinality().is_prime()
True
sage: B = E([
...     77150721626264982617064826856557988990776925417 6,
...     39015751024655662852527945926651499556253319665 5])
sage: n=67080503113991051351752720769306045630021705 4473
sage: r=70674630913457179596452846564371866229568459543
sage: P = E([14489646124220757767,
...     66933778037328409627489513661819460446969683007 4])
sage: encrypt = (r*B, P + r*(n*B))
sage: encrypt[1] - n*encrypt[0] == P   # decrypting works
True
```

6.4.3 타원곡선 이산로그 문제

문제 6.4.3 (타원곡선 이산로그 문제)**.** E가 $\mathbf{Z}/p\mathbf{Z}$위의 타원곡선이고 $P \in E(\mathbf{Z}/p\mathbf{Z})$는 E의 점이다. P의 배수 Q가 주어졌을 때, **타원곡선 이산로그 문제**(elliptic curve discrete log problem)는 $nP = Q$를 만족하는 $n \in \mathbf{Z}$을 구하는 문제이다.

예를 들어, 체 $\mathbf{Z}/7\mathbf{Z}$위에서 방정식 $y^2 = x^3 + x + 1$ 으로 정의된 타원곡선 E를 생각하자. 그러면 E는

$$E(\mathbf{Z}/7\mathbf{Z}) = \{\mathcal{O}, (2,2), (0,1), (0,6), (2,5)\}$$

인 원소가 5개인 군이다. 만약 $P = (2,2)$, $Q = (0,6)$이면, $3P = Q$이므로 $n = 3$이 이산로그문제의 답이다.

$E(\mathbf{Z}/p\mathbf{Z})$의 위수가 p이거나, $p \pm 1$이거나, 혹은 적당히 작은 소수들의 곱이면, E의 이산로그문제를 공격할 수 있는 방법이 있는데, 이 책의 범위를 벗어나므로 생략한다. 그러므로 명백한 취약성이 없는 암호체계를 만들기 위하여 타원곡선을 사용하는 경우에는 $\#E(\mathbf{Z}/p\mathbf{Z})$를 효율적으로 계산할 수 있는 것이 상당히 중요하다 $\#E(\mathbf{Z}/p\mathbf{Z})$를 계산하는 단순한 알고리듬은 각각의 값 $x \in \mathbf{Z}/p\mathbf{Z}$에 대하여 $x^3 + ax + b$가 법 p에서 완전수가 되는 경우를 모두 세는 것이다. 그러나 이 방법은 p가 암호에 사용할만큼 충분히 큰 경우는 쓸모없는 방법이다. 다행히도, Schoof, Elkies, 그리고 Atkin 덕분에, $\#E(\mathbf{Z}/p\mathbf{Z})$를 ($p$의 자릿수에 의존하는 다항식 시간안에)효율적으로 계산하는 알고리듬이 존재하지만 이 알고리듬도 우리 책의 범위를 넘어선다.

절 3.2.1에서, $(\mathbf{Z}/p\mathbf{Z})^*$에서의 이산로그문제를 설명했다. $(\mathbf{Z}/p\mathbf{Z})^*$에는 좀 느리긴 하지만, 그러나 일반적인 (추가적인 구조가 없는) 군의 경우보다는 더 빠른 'index calculus attacks"라고 불리는 일반적인 공격 방법이 있다. 대부분의 타원곡선에는 "index calculus attack"와 유사한 이산로그문제를 공격할 수 있는 알고리듬은 알려져 있지 않다. 현재로선 $E(\mathbf{Z}/p\mathbf{Z})$에서의 이산로그문제가 $(\mathbf{Z}/p\mathbf{Z})^*$에서의 이산로그문제보다 훨씬 더 어려운 것처럼 보인다. 또 타원곡선 암호는 훨씬 작은 수로 같은 수준의 보안을 유지하므로 $(\mathbf{Z}/p\mathbf{Z})^*$에 기반을 둔 암호보다는 타원곡선에 기반을 둔 암호체계를 사용하는 것을 권고하고 있으며, 이런 사실들이 일부 암호학자들에게 타원곡선 암호체계를 구축하는 것이 훨씬 매력적으로 보이는 이유이다. 예를 들어, 타원곡선 암호를 강하게 권고하는 회사인 Certicom의 주장이다:

> "[타원곡선암호] 도구는 다른 암호체계보다 더 적은 메모리 공간, 더 적은 전력, 그리고 더 적은 대역폭(bandwidth)을 요구한다. 이 때문에 무선장치, 초소형 컴퓨터, 스마트 카드, 얇은 장치 등과 같은 제한된 플랫폼에 암호를 구현할 수 있다. 또 효율성이 중요한 상황에서 암호를 성공적으로 구현할 수 있도록 한다. "

Certicom이 지원한 타원곡선 이산로그 도전 문제의 최신 목록들은 [7]에서 볼 수 있다. 그 중 하나는, 2004년 4월, $\mathbf{Z}/p\mathbf{Z}$위의 타원곡선에 기반한 암호가 깨졌는데, 이 때 소수 p는 109비트였다. 첫 번째 미해결 도전문제는 p기 131 비트 소수일 때 $\mathbf{Z}/p\mathbf{Z}$위의 타원곡선과 관련되어 있고, 그 다음 도전문제에서의

FIGURE 6.4. Louis J. Mordell

소수는 163 비트이다. Certicom은 [7]에서 163-비트 도전문제는 계산상으로는 실행불가능한 문제라고 주장한다.

6.5 유리수위에서의 타원곡선

E는 $\mathbf{Q}$ 위의 타원곡선이다. 다음은 군 $E(\mathbf{Q})$에 관한 아주 중요한 정리이다.

정리 6.5.1 (Mordell)**.** *군 $E(\mathbf{Q})$는 유한개의 원소로 생성된 군이다. 즉, 유한개의 점 $P_1, \ldots, P_s \in E(\mathbf{Q})$이 존재하여 $E(\mathbf{Q})$의 모든 점이 $n_1P_1 + \cdots + n_sP_s$ 형태로 나타난다. 이 때 $n_1, \ldots n_s \in \mathbf{Z}$이다.*

Mordell의 정리 덕에 $E(\mathbf{Q})$를 계산할 수 있는가를 질문하는 것이 의미가 있다. 여기서 계산한다는 것은 아벨군 $E(\mathbf{Q})$를 생성하는 E 위의 유한 집합 $P_1, \ldots, P_s$를 구하는 것이다. 실제 $E(\mathbf{Q})$를 구하는 "descent"라고 불리는 체계적인 접근법이 있다. ([14, 13, 48]참조). 이 "descent" 방법이 항상 성공할 것이라고 대부분이 믿지만 아직 아무도 증명하지는 못하였다. 이 "descent"가 모든 곡선들에 대하여 잘 작동할 것임을 증명하는 것은 정수론에서의 중요한 미해결 문제중의 하나이며 (Clay 수학 연구소의 백만달러 문제 중의 하나인) Birch와 Swinnerton-Dyer 가설과 밀접하게 연관되어 있다. 결정적인 어려움은 어떤 구체적으로 주어진 곡선이 (이 곡선들은 $\mathbf{R}$ 혹은 모든 법 n에서 정의되는 점들은 항상 가지는데) 유리수 점을 가지는 지 아닌지를 알아내는 것이다.

"descent" 방법을 이용하여 $E(\mathbf{Q})$를 계산하는 방법 또한 이 책의 범위를 넘어선다. 이제부터는 필요하다면 $E(\mathbf{Q})$는 어떤 구조를 가지거나 혹은 어떤 점들로부터 생성된다고 주장하려고 한다. 각 경우에서 우리는 이 방법의 컴퓨터 구현을 이용하여 $E(\mathbf{Q})$를 계산한다.

6.5.1 $E(\mathbf{Q})$의 꼬임부분군

G가 아벨군일 때 G의 꼬임부분군(torsion subgroup) G_{tor}는 유한 위수를 갖는 G의 원소들로 구성된 G의 부분군이다. E가 $\mathbf{Q}$ 위의 타원곡선이면, $E(\mathbf{Q})$의 부분군 $E(\mathbf{Q})_{\text{tor}}$는 정리 6.5.1에 의하여 유한군이다 (Exercise 6.6 참조). 또는 단사축소 준동형사상 $E(\mathbf{Q})_{\text{tor}} \hookrightarrow E(\mathbf{Z}/p\mathbf{Z})$을 정의하는 소수가 존재하고 $E(\mathbf{Z}/p\mathbf{Z})$가 유한하다는 사실을 이용하여 $E(\mathbf{Q})_{\text{tor}}$가 유한군임은 보일 수 있다. 예를 들어 E가 $y^2 = x^3 - 5x + 4$으로 정의되었다면, $E(\mathbf{Q})_{\text{tor}} = \{\mathcal{O}, (1,0)\} \cong \mathbf{Z}/2\mathbf{Z}$가 성립한다. $E(\mathbf{Q})_{\text{tor}}$는 어떤 형태여야 하는지는 알려져 있다.

정리 6.5.2 (Mazur, 1976)**.** *E가 $\mathbf{Q}$위의 타원곡선이면 $E(\mathbf{Q})_{\text{tor}}$는 다음 15개의 군들 중의 하나와 동형이다.*

$$\begin{aligned} &\mathbf{Z}/n\mathbf{Z} && (n \le 10 \text{ 혹은 } n = 12),\\ &\mathbf{Z}/2\mathbf{Z} \times \mathbf{Z}/2n && n \le 4. \end{aligned}$$

SAGE 예 6.5.3. 이 예에서는 타원곡선들의 꼬임부분군들을 구한다. 각 경우 아래의 함수 $T(a,b)$의 출력물은 $c, d \in \mathbf{Z}$인데, 이 출력물은 $y^3 = x^3 + ax + b$의 torsion 부분군 $\mathbf{Z}/c\mathbf{Z} \times \mathbf{Z}/d\mathbf{Z}$을 의미한다.

```
sage: T = lambda v: EllipticCurve(v
...           ).torsion_subgroup().invariants()
sage: T([-5,4])
(2,)
sage: T([-43,166])
(7,)
sage: T([-4,0])
(2, 2)
sage: T([-1386747, 368636886])
(2, 8)
```

6.5.2 $E(\mathbf{Q})$의 랭크

잉여군(factor group) $E(\mathbf{Q})/E(\mathbf{Q})_{\text{tor}}$은 유한 개의 원소로 생성되는 자유아벨군이므로, 적당한 자연수 r이 존재하여 $\mathbf{Z}^r$가 동형이다. 이 r을 $E(\mathbf{Q})$의 **랭크 (rank)**라 부른다. 예를 들어 E 가 $y^2 = x^3 - 5x + 4$으로 정의된 타원곡선이라면, $E(\mathbf{Q})/E(\mathbf{Q})_{\text{tor}}$는 $(0,2)$로 생성되었음을 보일 수 있다.

SAGE 예 6.5.4. Sage를 이용하여 타원곡선 $y^2 = x^3 + ax + b$의 랭크를 구한다. 아래에 정의하는 함수 $r(a,b)$는 $\mathbf{Q}$위의 타원곡선의 랭크를 알려준다.

```
sage: r = lambda v: EllipticCurve(v).rank()
sage: r([-5,4])
1
sage: r([0,1])
0
```

```
sage: r([-3024, 46224])
2
sage: r([-112, 400])
3
sage: r([-102627, 12560670])
4
```

다음 정리는 어떤 특정 수학자에게 속하지 않지만 대부분이 추측하는 가설이다.

가설 6.5.5. *임의의 큰 수가 랭크가 되는* $\mathbf{Q}$*위의 타원곡선이 존재한다.*

알려진 가장 큰 랭크는, 적어도 28이상으로 알려진, 다음 타원곡선의 랭크이다.

$$\begin{aligned} y^2+xy+y = x^3 - x^2- \\ 20067762415575526585033208209338542750930230312178956502x+ \\ 3448161179503055646703298569039072037485594435931918036 12\ldots \\ \ldots 66008296291939448732243429 \end{aligned}$$

이 예는 하버드 대학의 Noam Elkies가 2006년 5월에 발견하였다.

6.5.3 합동수 문제

정의 6.5.6 (합동수)**.** 영이 아닌 유리수 n의 절대값이 세 변의 길이가 유리수인 직각삼각형의 면적으로 나타나면 유리수 n을 **합동수(congruent number)**라고 부른다. 동치인 식으로 표현하면, $a, b, c \in \mathbf{Q}$인 다음 연립 방정식

$$\begin{aligned} a^2 + b^2 &= c^2 \\ \frac{1}{2}ab &= n \end{aligned}$$

이 유리수해 n을 가지면 n을 **합동수**(congruent number)라고 부른다.

예를 들어, 6은 직각을 낀 두 변의 길이가 각각 3, 4인 직각삼각형의 면적이므로 6은 합동수이다. 덜 분명하긴 하지만 5는 합동수이다: 세 변의 길이가 각각 3/2, 20/3, 41/6인 직각삼각형이 면적이다. 간단히 증명할 수는 없지만 1, 2, 3, 4는 합동수가 아니다. 다음은 50까지의 합동수의 목록이다.

$$5, 6, 7, 13, 14, 15, 20, 21, 22, 23, 24, 28, 29, 30, 31, 34, 37, 38, 39, 41, 45, 46, 47.$$

이 목록에는 3을 제외한 법 8의 모든 잉여류가 나타나고, 따라서 정확하지는 않지만 만약 $n \equiv 3 \pmod 8$이면 n은 합동수가 아니라는 것을 추론할 수 있다. 그러나 비록 $n \leq 218$까지는 $n \equiv 3 \pmod 8$인 n이 존재하지는 않지만 $n = 219$은 합동수이고, $219 \equiv 3 \pmod 8$이다.

면적이 n인 가장 단순한 삼각형이 굉장히 복잡할 수도 있으므로, 정수 n이 합동수인지를 결정하는 것은 감지하는 것은 쉽지 않다. 예를 들어 Zagier가 지적하였듯이, 157은 합동수이고 "가장 간단한" 유리수 길이를 갖는 직각삼각형의 직각을 낀 두 변의 길이는 다음과 같다.

$$a = \frac{6803298487826435051217540}{411340519227716149383203}, \quad b = \frac{41134051922771614938320 3}{21666555693714761309610}.$$

모든 가능한 경우를 샅샅이 찾아(brute force search) 열심히 계산함으로써 이 해를 찾는 것은 상당히 어려울 것이다.

합동수에 합동이란 용어를 쓴 이유는, 다음 기초정리에서 확인할 수 있듯이 임의의 함동수는 법 n에 합동인 세 제곱수와 관련되어 있기 때문이다.

기초정리 6.5.7. *정수 n이 세 변이 길이가 유리수 a, b, c인 직각삼각형의 면적이라고 가정하자. 단 $a \leq b < c$이다. $A = (c/2)^2$로 잡자. 그러면 두 정수*

$$A - n, \quad A, \quad A + n$$

도 유리수의 제곱이다.

증명 조건으로부터

$$\begin{aligned} a^2 + b^2 &= c^2 \\ \frac{1}{2}ab &= n \end{aligned}$$

이 성립한다. 두 번째 식에 4 를 곱하여 첫 번째 식에 더하거나 빼면

$$\begin{aligned} a^2 \pm 2ab + b^2 &= c^2 \pm 4n \\ (a \pm b)^2 &= c^2 \pm 4n \\ \left(\frac{a \pm b}{2}\right)^2 &= \left(\frac{c}{2}\right)^2 \pm n \\ &= A \pm n \end{aligned}$$

를 얻는다. □

주요한 동기부여를 하는 합동수에 관련된 미해결 문제는 합동수임을 알아낼 수 있는 체계적인 방법을 알아내는 것이다.

Open Problem 6.5.8. *유리수 n이 주어졌을 때, n이 합동수인지 아닌지를 출력하는 알고리듬을 찾아라.*

다행히도 타원곡선에 관한 광범위한 이론이 위의 문제에 대한 실마리를 준다. 타원곡선과 합동수와의 관계를 이해하기 위하여, 기초 대수학적 기초정리로 시작한다.

기초정리 6.5.9 (합동수와 타원곡선)**.** *n은 유리수이다. 그러면 다음 두 집합 A와 B사이에 일대일 대응관계가 존재한다.*

$$A = \left\{ (a,b,c) \in \mathbf{Q}^3 \,:\, \frac{ab}{2} = n,\, a^2+b^2 = c^2 \right\};$$

$$B = \left\{ (x,y) \in \mathbf{Q}^2 \,:\, y^2 = x^3 - n^2 x, \text{ with } y \neq 0 \right\}.$$

이 때 대응관계는 다음 두 함수로 정의된다.

$$f(a,b,c) = \left(-\frac{nb}{a+c},\ \frac{2n^2}{a+c} \right),$$

$$g(x,y) = \left(\frac{n^2-x^2}{y},\ -\frac{2xn}{y},\ \frac{n^2+x^2}{y} \right).$$

이 기초정리의 증명은 심오하지는 않지만, 이 책에서는 증명하지 않은 상당한 양의 기초 대수학 지식이 필요하다.

$n \neq 0$일 때, E_n은 $y^2 = x^3 - n^2x$로 정의된 타원곡선이다.

기초정리 6.5.10 (합동수 판정))**.** *유리수 n이 합동수이기 위한 필요충분조건은 $E_n(\mathbf{Q})$에 $y \neq 0$인 점이 하나 존재하는 것이다.*

증명 n이 합동수라는 것은 정의에 의하여 기초정리 6.5.9의 집합 A가 공집합이 아니다. 기초정리 6.5.9에 의하여 집합 A와 집합 B가 일대일 대응이므로 주장이 성립한다. □

예 6.5.11. $n = 5$로 놓자. 그러면 E_n은 $y^2 = x^3 - 25x$으로 정의되므로, $(-4,-6) \in E_n(\mathbf{Q})$임을 확인할 수 있다. 이제 기초정리 6.5.9의 일대일 대응함수를 이용하여 대응하는 직각삼각형을 찾는다.

$$g(-4,-6) = \left(\frac{25-16}{-6}, -\frac{-40}{-6}, \frac{25+16}{-6} \right) = \left(-\frac{3}{2}, -\frac{20}{3}, -\frac{41}{6} \right).$$

-1을 곱함으로써, 면적이 5인 직각삼각형의 세 변의 길이를 얻는다. *다른 것들도 있나요?*

기초정리 6.5.9의 함수 g에는 $y \neq 0$인 $E_n(\mathbf{Q})$의 모든 점에 g를 적용할 수 있다. 타원군의 덧셈 연산을 이용하면 $2(-4,-6) = (1681/144, 62279/1728)$이므로,

$$g(2(-4,-6)) = \left(-\frac{1519}{492}, -\frac{4920}{1519}, \frac{3344161}{747348} \right)$$

를 얻을 수 있다. 이 예가 다음 정리 6.5.14를 예고한다.

예 6.5.12. $n = 1$이면, E_1은 $y^2 = x^3 - x$으로 정의된다. 1은 합동수가 아니므로 타원곡선 E_1은 $y \neq 0$인 점이 없다. Exercise 6.11 참조.

SAGE 예 6.5.13. 정수 n을 주고, 존재하는 경우, 면적이 n인 유리수 길이를 갖는 직각삼각형의 세 변의 길이가 되는 세 쌍의 유리수 (a,b,c)를 출력하고, 존재하지 않으면 False 를 출력하는 Sage 함수 `cong`를 구현한다.

```
sage: def cong(n):
...       G = EllipticCurve([-n^2,0]).gens()
...       if len(G) == 0: return False
...       x,y,_ = G[0]
...       return ((n^2-x^2)/y,-2*x*n/y,(n^2+x^2)/y)
sage: cong(6)
(3, 4, 5)
sage: cong(5)
(3/2, 20/3, 41/6)
sage: cong(1)
False
sage: cong(13)
(323/30, 780/323, 106921/9690)
sage: (323/30 * 780/323)/2
13
sage: (323/30)^2 + (780/323)^2 == (106921/9690)^2
True
```

정리 6.5.14 (무한히 많은 삼각형들)**.** *만약 n이 합동수이면, 세 변의 길이가 유리수이면서 면적이 n이 되는 직각 삼각형이 무한히 많이 존재한다.*

이 정리는 다 증명하지는 않는다. 그러나 $E_n(\mathbf{Q})$의 위수가 유한인 원소들의 부분집합 $E_n(\mathbf{Q})_{\text{tor}}$가 $E_n(\mathbf{Q})_{\text{tor}} = \{\mathcal{O}, (0,0), (n,0), (-n,0)\}$임은 언급하고자 한다. 그러면 기초정리 6.5.9의 집합 B의 모든 원소는 위수가 무한이다. 따라서 B는 무한집합이 되고 따라서 A도 무한집합이다.

Tunnell은 (앞서 언급한) Birch와 Swinnerton-Dyer 가설이 정수 n이 합동수인지 아닌지를 결정할 수 있는 쉬운 방법이 존재하는 것을 내포하고 있음을 증명하였다. Tunnell의 기본 밥법을 가설의 형태로 소개한다.

가설 6.5.15. *a, b, c는 정수이다. 만약 n이 제곱수 약수가 없는 짝수라면, n이 합동수이기 위한 필요충분조건은*

$$\#\left\{(a,b,c) \in \mathbf{Z}^3 : 4a^2 + b^2 + 8c^2 = \frac{n}{2} : c \text{ is even}\right\}$$
$$= \#\left\{(a,b,c) : 4a^2 + b^2 + 8c^2 = \frac{n}{2} : c \text{ is odd}\right\}$$

이다. 만약 n이 제곱수 약수가 없는 홀수라면, n이 합동수이기 위한 필요충분조건은

$$\#\left\{(a,b,c) : 2a^2 + b^2 + 8c^2 = n : c \text{ is even}\right\}$$
$$= \#\left\{(a,b,c) : 2a^2 + b^2 + 8c^2 = n : c \text{ is odd}\right\}$$

이다.

Birch와 Swinnerton-Dyer 가설에 대해 가설 6.5.15의 한 방향을 증명할 수 있을 만큼은 알려져 있다. 아주 어려운 정리이긴 하지만, 위에 나타낸 두 집합의 크기가 다르면 n은 합동수가 아니라는 것은 증명되었다.

가설 6.5.15에서 한층 더 어려운 (여전히 미해결 상태인) 부분은 위의 역인 가설 6.5.15의 두 집합의 크기가 같으면, n이 합동수임을 보이는 것이다. 이 방향을 증명할 때 어려운 문제는 어떻게든 $E_n(\mathbf{Q})$의 원소를 찾아야 하는데, 이 부분이 아주 깊이 있는 연구가 필요할 것으로 예상되는 부분이다. Gross와 Zagier([20])의 획기적인 연구 덕에 몇 경우는 해결되었지만 여전히 해야 할 연구들이 많이 남아 있다.

합동수와 가설 6.5.15에 대해서는 아주 잘 쓰여진 책으로 알려진 [29]를 참고하기를 추천한다. Birch와 Swinnerton-Dyer 가설은 클레이 수학 연구소(Clay Math Institute)의 백만불짜리 문제 중의 하나이다. (see [8, 54])

6.6 Exercises

6.1 Write down an equation $y^2 = x^3 + ax + b$ over a field K such that $-16(4a^3+27b^2) = 0$. Precisely what goes wrong when trying to endow the set $E(K) = \{(x, y) \in K \times K : y^2 = x^3 + ax + b\} \cup \{\mathcal{O}\}$ with a group structure?

6.2 One rational solution to the equation $y^2 = x^3 - 2$ is $(3, 5)$. Find a rational solution with $x \neq 3$ by drawing the tangent line to $(3, 5)$ and computing the second point of intersection.

6.3 Let E be the elliptic curve over the finite field $K = \mathbf{Z}/5\mathbf{Z}$ defined by the equation

$$y^2 = x^3 + x + 1.$$

(a) List all 9 elements of $E(K)$.

(b) What is the structure of $E(K)$, as a product of cyclic groups?

6.4 Let E be the elliptic curve defined by the equation $y^2 = x^3 + 1$. For each prime $p \geq 5$, let N_p be the cardinality of the group $E(\mathbf{Z}/p\mathbf{Z})$ of points on this curve having coordinates in $\mathbf{Z}/p\mathbf{Z}$. For example, we have that $N_5 = 6, N_7 = 12, N_{11} = 12, N_{13} = 12, N_{17} = 18, N_{19} = 12, , N_{23} = 24,$ and $N_{29} = 30$ (you do not have to prove this).

(a) For the set of primes satisfying $p \equiv 2 \pmod 3$, can you see a pattern for the values of N_p? Make a general conjecture for the value of N_p when $p \equiv 2 \pmod 3$.

(b) (*) Prove your conjecture.

6.5 Let E be an elliptic curve over the real numbers $\mathbf{R}$. Prove that $E(\mathbf{R})$ is not a finitely generated abelian group.

6.6 (*) Suppose G is a finitely generated abelian group. Prove that the subgroup G_{tor} of elements of finite order in G is finite.

6.7 Suppose $y^2 = x^3 + ax + b$ with $a, b \in \mathbf{Q}$ defines an elliptic curve. Show that there is another equation $Y^2 = X^3 + AX + B$ with $A, B \in \mathbf{Z}$ whose solutions are in bijection with the solutions to $y^2 = x^3 + ax + b$.

6.8 Suppose a, b, c are relatively prime integers with $a^2 + b^2 = c^2$. Then there exist integers x and y with $x > y$ such that $c = x^2 + y^2$ and either $a = x^2 - y^2$, $b = 2xy$ or $a = 2xy$, $b = x^2 - y^2$.

6.9 (*) Fermat's Last Theorem for exponent 4 asserts that any solution to the equation $x^4 + y^4 = z^4$ with $x, y, z \in \mathbf{Z}$ satisfies $xyz = 0$. Prove Fermat's Last Theorem for exponent 4, as follows.

(a) Show that if the equation $x^2 + y^4 = z^4$ has no integer solutions with $xyz \neq 0$, then Fermat's Last Theorem for exponent 4 is true.

(b) Prove that $x^2 + y^4 = z^4$ has no integer solutions with $xyz \neq 0$ as follows. Suppose $n^2 + k^4 = m^4$ is a solution with $m > 0$ minimal among all solutions. Show that there exists a solution with m smaller using Exercise 6.8 (consider two cases).

6.10 This problem requires a computer.

(a) Show that the set of numbers $59 + 1 \pm s$ for $s \leq 15$ contains 14 numbers that are B-power smooth for $B = 20$.

(b) Find the proportion of primes p in the interval from 10^{12} and $10^{12} + 1000$ such that $p - 1$ is $B = 10^5$ power smooth.

6.11 (*) Prove that 1 is not a congruent number by showing that the elliptic curve $y^2 = x^3 - x$ has no rational solutions except $(0, \pm 1)$ and $(0, 0)$, as follows:

(a) Write $y = \frac{p}{q}$ and $x = \frac{r}{s}$, where p, q, r, s are all positive integers and $\gcd(p, q) = \gcd(r, s) = 1$. Prove that $s \mid q$, so $q = sk$ for some $k \in \mathbf{Z}$.

(b) Prove that $s = k^2$, and substitute to see that $p^2 = r^3 - rk^4$.

(c) Prove that r is a perfect square by supposing that there is a prime ℓ such that $\mathrm{ord}_\ell(r)$ is odd, and analyzing ord_ℓ of both sides of $p^2 = r^3 - rk^4$.

(d) Write $r = m^2$, and substitute to see that $p^2 = m^6 - m^2k^4$. Prove that $m \mid p$.

(e) Divide through by m^2 and deduce a contradiction to Exercise 6.9.

Answers and Hints

- **Chapter 1. Prime Numbers**

  2. They are $2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97$.
  3. Emulate the proof of Proposition 1.2.5.

- **Chapter 2. The Ring of Integers Modulo n**

  2. They are 5, 13, 3, and 8.
  3. For example, $x = 22$, $y = -39$.
  4. Hint: Use the binomial theorem and prove that if $r \geq 1$, then p divides $\binom{p}{r}$.
  7. For example, $S_1 = \{0, 1, 2, 3, 4, 5, 6\}$, $S_2 = \{1, 3, 5, 7, 9, 11, 13\}$, $S_3 = \{0, 2, 4, 6, 8, 10, 12\}$, and $S_4 = \{2, 3, 5, 7, 11, 13, 29\}$. In each we find S_i by listing the first seven numbers satisfying the ith condition, then adjust the last number if necessary so that the reductions will be distinct modulo 7.
  8. An integer is divisible by 5 if and only if the last digits is 0 or 5. An integer is divisible by 9 if and only if the sum of the digits is divisible by 9. An integer is divisible by 11 if and only if the alternating sum of the digits is divisible by 11.

9. Hint for part (a): Use the divisibility rule you found in Exercise 1.8.

10. 71

11. 8

12. As explained on page 23, we know that $\mathbf{Z}/n\mathbf{Z}$ is a ring for any n. Thus to show that $\mathbf{Z}/p\mathbf{Z}$ is a field it suffices to show that every nonzero element $\overline{a} \in \mathbf{Z}/p\mathbf{Z}$ has an inverse. Lift a to an element $a \in \mathbf{Z}$, and set $b = p$ in Proposition 2.3.1. Because p is prime, $\gcd(a, p) = 1$, so there exists x, y such that $ax+py = 1$. Reducing this equality modulo p proves that $\overline{a}$ has an inverse $x \pmod{p}$. Alternatively, one could argue just like after Definition 2.1.16 that $\overline{a}^m = 1$ for some m, so some power of $\overline{a}$ is the inverse of $\overline{a}$.

13. 302

15. Only for $n = 1, 2$. If $n > 2$, then n is either divisible by an odd prime p or 4. If $4 \mid n$, then $2^e - 2^{e-1}$ divides $\varphi(n)$ for some $e \geq 2$, so $\varphi(n)$ is even. If an odd p divides n, then the even number $p^e - p^{e-1}$ divides $\varphi(n)$ for some $e \geq 1$.

16. The map ψ is a homomorphism since both reduction maps

$$\mathbf{Z}/mn\mathbf{Z} \to \mathbf{Z}/m\mathbf{Z} \quad \text{and} \quad \mathbf{Z}/mn\mathbf{Z} \to \mathbf{Z}/n\mathbf{Z}$$

are homomorphisms. It is injective because if $a \in \mathbf{Z}$ is such that $\psi(a) = 0$, then $m \mid a$ and $n \mid a$, so $mn \mid a$ (since m and n are coprime), so $a \equiv 0 \pmod{mn}$. The cardinality of $\mathbf{Z}/mn\mathbf{Z}$ is mn and the cardinality of the product $\mathbf{Z}/m\mathbf{Z} \times \mathbf{Z}/n\mathbf{Z}$ is also mn, so ψ must be an isomorphism. The units $(\mathbf{Z}/mn\mathbf{Z})^*$ are thus in bijection with the units $(\mathbf{Z}/m\mathbf{Z})^* \times (\mathbf{Z}/n\mathbf{Z})^*$.

For the second part of the exercise, let $g = \gcd(m, n)$ and set $a = mn/g$. Then $a \not\equiv 0 \pmod{mn}$, but $m \mid a$ and $n \mid a$, so $a \ker(\psi)$.

17. We express the question as a system of linear equations modulo various numbers, and use the Chinese remainder theorem. Let x be the number of books. The problem asserts that

$$\begin{aligned} x &\equiv 6 \pmod{7} \\ x &\equiv 2 \pmod{6} \\ x &\equiv 1 \pmod{5} \\ x &\equiv 0 \pmod{4} \end{aligned}$$

Applying CRT to the first pair of equations, we find that $x \equiv 20 \pmod{42}$. Applying CRT to this equation and the third, we find that $x \equiv 146 \pmod{210}$. Since 146 is not divisible by 4, we add

multiples of 210 to 146 until we find the first x that is divisible by 4. The first multiple works, and we find that the aspiring mathematicians have 356 math books.

18. Note that $p = 3$ works, since $11 = 3^2 + 2$ is prime. Now suppose $p \neq 3$ is any prime such that p and $p^2 + 2$ are both prime. We must have $p \equiv 1 \pmod 3$ or $p \equiv 2 \pmod 3$. Then $p^2 \equiv 1 \pmod 3$, so $p^2 + 2 \equiv 0 \pmod 3$. Since $p^2 + 2$ is prime, we must have $p^2 + 2 = 3$, so $p = 1$, a contradiction as p is assumed prime.

19. For (a) $n = 1, 2$, see solution to Exercise 2.15. For (b), yes there are many such examples. For example, $m = 2$, $n = 4$.

20. By repeated application of multiplicativity and Equation (2.2.2) on page 32, we see that if $n = \prod_i p_i^{e_i}$ is the prime factorization of n, then

$$\varphi(n) = \prod_i (p_i^{e_i} - p_i^{e_i - 1}) = \prod_i p_i^{e_i - 1} \cdot \prod_i (p_i - 1).$$

23. 1, 6, 29, 34

24. Let $g = \gcd(12n+1, 30n+2)$. Then $g \mid 30n+2-2\cdot(12n+1) = 6n$. For the same reason, g also divides $12n + 1 - 2 \cdot (6n) = 1$, so $g = 1$, as claimed.

27. There is no primitive root modulo 8, since $(\mathbf{Z}/8\mathbf{Z})^*$ has order 4, but every element of $(\mathbf{Z}/8\mathbf{Z})^*$ has order 2. Prove that if ζ is a primitive root modulo 2^n, for $n \geq 3$, then the reduction of ζ mod 8 is a primitive root, a contradiction.

28. 2 is a primitive root modulo 125.

29. Let $\prod_{i=1}^m p_i^{e_i}$ be the prime factorization of n. Slightly generalizing Exercise 16, we see that

$$(\mathbf{Z}/n\mathbf{Z})^* \cong \prod (\mathbf{Z}/p_i^{e_i}\mathbf{Z})^*.$$

Thus $(\mathbf{Z}/n\mathbf{Z})^*$ is cyclic if and only if the product $(\mathbf{Z}/p_i^{e_i}\mathbf{Z})^*$ is cyclic. If $8 \mid n$, then there is no chance $(\mathbf{Z}/n\mathbf{Z})^*$ is cyclic, so assume $8 \nmid n$. Then by Exercise 2.28, each group $(\mathbf{Z}/p_i^{e_i}\mathbf{Z})^*$ is itself cyclic. A product of cyclic groups is cyclic if and only the orders of the factors in the product are coprime (this follows from Exercise 2.16). Thus $(\mathbf{Z}/n\mathbf{Z})^*$ is cyclic if and only if the numbers $p_i(p_i - 1)$, for $i = 1, \ldots, m$ are pairwise coprime. Since $p_i - 1$ is even, there can be at most one odd prime in the factorization of n, and we see that $(\mathbf{Z}/n\mathbf{Z})^*$ is cyclic if and only if n is an odd prime power, twice an odd prime power, or $n = 4$.

• Chapter 3. Public-Key Cryptography

1. The best case is that each letter is A. Then the question is to find the largest n such that $1+27+\cdots+27^n \le 10^{20}$. By computing $\log_{27}(10^{20})$, we see that $27^{13} < 10^{20}$ and $27^{14} > 10^{20}$. Thus $n \le 13$, and since $1+27+\cdots+27^{n-1} < 27^n$, and $2\cdot 27^{13} < 10^{20}$, it follows that $n = 13$.
2. This is not secure, since it is just equivalent to a "Ceaser Cipher," that is a permutation of the letters of the alphabet, which is well-known to be easily broken using a frequency analysis.
3. If we can compute the polynomial

   $$f = (x-p)(x-q)(x-r) = x^3-(p+q+r)x^2+(pq+pr+qr)x-pqr,$$

   then we can factor n by finding the roots of f, for example, using Newton's method (or Cardona's formula for the roots of a cubic). Because p, q, r, are distinct odd primes, we have

   $$\varphi(n) = (p-1)(q-1)(r-1) = pqr-(pq+pr+qr)+p+q+r,$$

   and

   $$\sigma(n) = 1+(p+q+r)+(pq+pr+qr)+pqr.$$

   Since we know n, $\varphi(n)$, and $\sigma(n)$, we know

   $$\sigma(n)-1-n = (p+q+r)+(pq+pr+qr), \quad \text{and}$$
   $$\varphi(n)-n = (p+q+r)-(pq+pr+qr).$$

   We can thus compute both $p+q+r$ and $pq+pr+qr$, hence deduce f and find p, q, r.

• Chapter 4. Quadratic Reciprocity

1. They are all 1, -1, 0, and 1.
3. By Proposition 4.3.4, the value of $\left(\frac{3}{p}\right)$ depends only on the reduction $\pm p \pmod{12}$. List enough primes p such that $\pm p$ reduce to $1, 5, 7, 11$ modulo 12 and verify that the asserted formula holds for each of them.
7. Since $p = 2^{13}-1$ is prime, there are either two solutions or no solutions to $x^2 \equiv 5 \pmod{p}$, and we can decide which using quadratic reciprocity. We have

   $$\left(\frac{5}{p}\right) = (-1)^{(p-1)/2\cdot(5-1)/2}\left(\frac{p}{5}\right) = \left(\frac{p}{5}\right),$$

   so there are two solutions if and only if $p = 2^{13}-1$ is ± 1 mod 5. In fact, $p \equiv 1 \pmod{5}$, so there are two solutions.

8. We have $4^{48} = 2^{96}$. By Euler's Theorem, $2^{96} = 1$, so $x = 1$.

9. For (a), take $a = 19$ and $n = 20$. We found this example using the Chinese remainder theorem applied to $4 \pmod 5$ and $3 \pmod 4$, and used that $\left(\frac{19}{20}\right) = \left(\frac{19}{5}\right) \cdot \left(\frac{19}{4}\right) = (-1)(-1) = 1$, yet 19 is not a square modulo either 5 or 4, so is certainly not a square modulo 20.

10. Hint: First reduce to the case that $6k - 1$ is prime, by using that if p and q are primes not of the form $6k - 1$, then neither is their product. If $p = 6k - 1$ divides $n^2 + n + 1$, it divides $4n^2 + 4n + 4 = (2n + 1)^2 + 3$, so -3 is a quadratic residue modulo p. Now use quadratic reciprocity to show that -3 is not a quadratic residue modulo p.

• Chapter 5. Continued Fractions

9. Suppose $n = x^2 + y^2$, with $x, y \in \mathbf{Q}$. Let d be such that $dx, dy \in \mathbf{Z}$. Then $d^2n = (dx)^2 + (dy)^2$ is a sum of two integer squares, so by Theorem 5.7.1, if $p \mid d^2n$ and $p \equiv 3 \pmod 4$, then $\mathrm{ord}_p(d^2n)$ is even. We have $\mathrm{ord}_p(d^2n)$ is even if and only if $\mathrm{ord}_p(n)$ is even, so Theorem 5.7.1 implies that n is also a sum of two squares.

11. The squares modulo 8 are $0, 1, 4$, so a sum of two squares reduces modulo 8 to one of $0, 1, 2, 4$, or 5. Four consecutive integers that are sums of squares would reduce to four consecutive integers in the set $\{0, 1, 2, 4, 5\}$, which is impossible.

• Chapter 6. Elliptic Curves

2. The second point of intersection is $(129/100, 383/1000)$.

3. The group is cyclic of order 9, generated by $(4, 2)$. The elements of $E(K)$ are

$$\{\mathcal{O}, (4,2), (3,4), (2,4), (0,4), (0,1), (2,1), (3,1), (4,3)\}.$$

4. In part (a), the pattern is that $N_p = p + 1$. For part (b), a hint is that when $p \equiv 2 \pmod 3$, the map $x \mapsto x^3$ on $(\mathbf{Z}/p\mathbf{Z})^*$ is an automorphism, so $x \mapsto x^3 + 1$ is a bijection. Now use what you learned about squares in $\mathbf{Z}/p\mathbf{Z}$ from Chapter 4.

5. For all sufficiently large real x, the equation $y^2 = x^3 + ax + b$ has a real solution y. Thus, the group $E(\mathbf{R})$ is not countable, since $\mathbf{R}$ is not countable. But any finitely generated group is countable.

6. In a course on abstract algebra, one often proves the nontrivial fact that every subgroup of a finitely generated abelian group is finitely generated. In particular, the torsion subgroup G_{tor} is finitely generated. However, a finitely generated abelian torsion group is finite.
7. Hint: Multiply both sides of $y^2 = x^3 + ax + b$ by a power of a common denominator, and "absorb" powers into x and y.
8. Hint: see Exercise 4.6.

References

[1] [ACD+99] K. Aardal, S. Cavallar, B. Dodson, A. Lenstra, W. Lioen, P. L. Montgomery, B. Murphy, J. Gilchrist, G. Guillerm, P. Leyland, J. Marchand, F. Morain, A. Muffett, C.&C. Putnam, and P. Zimmermann, *Factorization of a 512-bit RSA key using the Number Field Sieve*, http://www.loria.fr/ zimmerma/records/RSA155 (1999).

[2] W. R. Alford, Andrew Granville, and Carl Pomerance, *There are infinitely many Carmichael numbers*, Ann. of Math. (2) 139 (1994), no. 3, 703-722. MR 95k:11114

[3] M. Agrawal, N. Kayal, and N. Saxena, *PRIMES is in P*, to appear in Annals of Math., http://www.cse.iitk.ac.in/users/manindra/primality.ps (2002).

[4] Leonard E. Baum and Melvin M. Sweet, *Continued fractions of algebraic power series in characteristic 2*, Ann. of Math. (2) 103 (1976), no. 3, 593-610. MR 53 #13127

[5] D. M. Burton, *Elementary Number Theory*, second ed., W. C. Brown Publishers, Dubuque, IA, 1989. MR 90e:11001

[6] C. Caldwell, *The Largest Known Primes*, http://www.utm.edu/research/primes/largest.html. 156 References

[7] Certicom, *The certicom ECC challenge*, http://www.certicom.com/ index.php?action=res,ecc challenge.

[8] Clay Mathematics Institute, *Millennium prize problems*, http://www.claymath.org/millennium prize problems/.

[9] H. Cohn, *A short proof of the continued fraction expansion of e*, http://research.microsoft.com/ cohn/publications.html.

[10] H. Cohen, *A Course in Computational Algebraic Number Theory*, Graduate Texts in Mathematics, vol. 138, Springer-Verlag, Berlin, 1993. MR 94i:11105

[11] John H. Conway, *The Sensual (Quadratic) Form*, Carus Mathematical Monographs, vol. 26, Mathematical Association of America, Washington, DC, 1997, With the assistance of Francis Y. C. Fung. MR 98k:11035

[12] R. Crandall and C. Pomerance, *Prime Numbers*, Springer-Verlag, New York, 2001, A computational perspective. MR 2002a:11007

[13] J. E. Cremona, mwrank *(computer software)*, http://www.maths.nott.ac.uk/personal/jec/ftp/progs/.

[14] ____, *Algorithms for modular elliptic curves*, second ed., Cambridge University Press, Cambridge, 1997.

[15] H. Davenport, *The Higher Arithmetic*, seventh ed., Cambridge University Press, Cambridge, 1999, An introduction to the theory of numbers, Chapter VIII by J. H. Davenport. MR 2000k:11002

[16] W. Diffie and M. E. Hellman, *New directions in cryptography*, IEEE Trans. Information Theory IT-22 (1976), no. 6, 644-654. MR 55 #10141

[17] Leonhard Euler, *An essay on continued fractions*, Math. Systems Theory 18 (1985), no. 4, 295-328, Translated from the Latin by B. F. Wyman and M. F. Wyman. MR 87d:01011b

[18] A. Frohlich and M. J. Taylor, *Algebraic Number Theory*, Cambridge University Press, Cambridge, 1993. MR 94d:11078

[19] R. K. Guy, Unsolved Problems in Number Theory, second ed., Springer-Verlag, New York, 1994, Unsolved Problems in Intuitive Mathematics, I. MR 96e:11002

[20] B. Gross and D. Zagier, *Heegner points and derivatives of L- series*, Invent. Math. 84 (1986), no. 2, 225-320. MR 87j:11057 References 157

[21] R. Hartshorne, *Algebraic Geometry*, Springer-Verlag, New York, 1977, Graduate Texts in Mathematics, No. 52.

[22] C. Hooley, *On Artin's conjecture*, J. Reine Angew. Math. 225 (1967), 209-220. MR 34 #7445

[23] G. H. Hardy and E. M. Wright, *An Introduction to the Theory of Numbers*, fifth ed., The Clarendon Press Oxford University Press, New York, 1979. MR 81i:10002

[24] IBM, *IBM's Test-Tube Quantum Computer Makes History*, http://www.research.ibm.com/resources/news/ 20011219 quantum.shtml.

[25] K. Ireland and M. Rosen, *A Classical Introduction to Modern Number Theory*, second ed., Springer-Verlag, New York, 1990. MR 92e:11001

[26] A. Ya. Khintchine, *Continued fractions*, Translated by Peter Wynn, P. Noordhoff Ltd., Groningen, 1963. MR 28 #5038

[27] Donald E. Knuth, *The Art of Computer Programming*, third ed., Addison-Wesley Publishing Co., Reading, Mass.-London- Amsterdam, 1997, Volume 1: Fundamental algorithms, Addison- Wesley Series in Computer Science and Information Processing.

[28] ____, *The Art of Computer Programming. Vol. 2*, second ed., Addison-Wesley Publishing Co., Reading, Mass., 1998, Seminumerical algorithms, Addison-Wesley Series in Computer Science and Information Processing. MR 83i:68003

[29] N. Koblitz, *Introduction to Elliptic Curves and Modular Forms*, Graduate Texts in Mathematics, vol. 97, Springer-Verlag, New York, 1984. MR 86c:11040

[30] D. N. Lehmer, *List of Primes Numbers from 1 to 10;006;721*, Carnegie Institution Washington, D.C. (1914).

[31] F. Lemmermeyer, *Proofs of the Quadratic Reciprocity Law*, http://www.rzuser.uni-heidelberg.de/ hb3/rchrono.html.

[32] H. W. Lenstra, Jr., *Factoring integers with elliptic curves*, Ann. of Math. (2) 126 (1987), no. 3, 649-673. MR 89g:11125

[33] A. K. Lenstra and H. W. Lenstra, Jr. (eds.), *The Development of the Number Field Sieve*, Lecture Notes in Mathematics, vol. 1554, Springer-Verlag, Berlin, 1993. MR 96m:11116 158 References

[34] Vandersypen L. M., Steffen M., Breyta G., Yannoni C. S., Sherwood M. H., and Chuang I. L., *Experimental realization of Shor's quantum factoring algorithm using nuclear magnetic resonance*, Nature 414 (2001), no. 6866, 883-887.

[35] S. Lang and H. Trotter, *Continued fractions for some algebraic numbers*, J. Reine Angew. Math. 255 (1972), 112-134; addendum, ibid. 267 (1974), 219-220; MR 50 #2086. MR 46 #5258

[36] ____, Addendum to: Continued fractions for some algebraic numbers (J. Reine Angew. Math. 255 (1972), 112-134), J. Reine Angew. Math. 267 (1974), 219-220. MR 50 #2086

[37] P. Moree, *A note on Artin's conjecture*, Simon Stevin 67 (1993), no. 3-4, 255-257. MR 95e:11106

[38] B. Mazur and W. Stein, *What is Riemann's Hypothesis?*, 2008, In preparation.

[39] I. Niven, H. S. Zuckerman, and H. L. Montgomery, *An Introduction to the Theory of Numbers*, fifth ed., John Wiley & Sons Inc., New York, 1991. MR 91i:11001

[40] C. D. Olds, *The Simple Continued Fraction Expression of e*, Amer. Math. Monthly 77 (1970), 968-974.

[41] O. Perron, *Die Lehre von den Kettenbruchen. Dritte, verbesserte und erweiterte Au . Bd. II. Analytisch-funktionentheoretische Kettenbruche*, B. G. Teubner Verlagsgesellschaft, Stuttgart, 1957. MR 19,25c

[42] RSA, *The New RSA Factoring Challenge*, http://www.rsasecurity.com/rsalabs/challenges/factoring.

[43] R. L. Rivest, A. Shamir, and L. Adleman, *A method for obtaining digital signatures and public-key cryptosystems*, Comm. ACM 21 (1978), no. 2, 120-126. MR 83m:94003

[44] Sage, *Free Open Source Mathematical Software (Version 3.0.4)*, 2008, http://www.sagemath.org.

[45] R. Schoof, *Elliptic curves over finite fields and the computation of square roots mod p*, Mathematics of Computation 44 (1985), no. 170, 483-494.

[46] P. W. Shor, *Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer*, SIAM J. Comput. 26 (1997), no. 5, 1484-1509. MR 98i:11108 References 159

[47] V. Shoup, *A Computational Introduction to Number Theory and Algebra*, Cambridge University Press, Cambridge, 2005. MR MR2151586 (2006g:11003)

[48] J. H. Silverman, *The arithmetic of elliptic curves*, Graduate Texts in Mathematics, vol. 106, Springer-Verlag, New York, 1986. MR 87g:11070

[49] S. Singh, *The Code Book: The Science of Secrecy from Ancient Egypt to Quantum Cryptography*, Doubleday, 1999.

[50] N. J. A. Sloane, *The On-Line Encyclopedia of Integer Sequences*, http://www.research.att.com/ njas/sequences/.

[51] J. H. Silverman and J. Tate, *Rational points on elliptic curves*, Undergraduate Texts in Mathematics, Springer-Verlag, New York, 1992. MR 93g:11003

[52] H. S. Wall, *Analytic Theory of Continued Fractions*, D. Van Nostrand Company, Inc., New York, N. Y., 1948. MR 10,32d

[53] E. W. Weisstein, *RSA-576 Factored*, http://mathworld.wolfram.com/news/2003-12-05/rsa/.

[54] A. J. Wiles, *The Birch and Swinnerton-Dyer Conjecture*, http://www.claymath.org/prize problems/birchsd.htm.

[55] D. Zagier, *The first 50 million prime numbers*, http://modular.fas.harvard.edu/scans/papers/zagier/.

Index