

**ALGORYTM** (średniowieczno-łac. *algorithmus*, od przydomka perskiego matematyka Abu Ćafara Muḥammada Ibn Mūsà al-Ḥuwārizmī zw. Al-Chwarizmi, zmieniony pod wpływem gr. na *arithmos*) — dokładny opis postępowania gwarantujący osiągnięcie określonego rezultatu w skończonej ilości elementarnych kroków.

Najistotniejsze cechy a.: 1) określona jest sytuacja początkowa, w której można a. zastosować, sytuacja docelowa oraz rodzaje czynności, jakie mogą być wykonywane; 2) wykonywane elementarne czynności są całkowicie jednoznaczne i możliwe do zrealizowania przez określonego wykonawcę a. Możliwość konstruowania a. przeznaczonych dla różnych wykonawców sprawia, że kwestia, czy dany przepis jest a., staje się relatywna — przy jednym wykonawcy przepis jest a., przy innym nie jest. Czasami żąda się, aby wykonanie poszczególnych kroków było czysto mechaniczne, tzn. nie wymagające rozumienia sensu czynności; 3) po wykonaniu kolejnego kroku przepisy wskazują, jaki ma być krok następny, bądź informują, że zadanie zostało wykonane. Przepis może przybierać postać instrukcji warunkowej: jeśli spełniony jest określony warunek, to wykonaj pewne działanie; 4) osiągnięcie pożądanego rezultatu jest gwarantowane — niezależnie od parametrów sytuacji początkowej wykonanie a. kończy się osiągnięciem sytuacji docelowej. Może się jednak zdarzyć, iż a. jest praktycznie niewykonalny, gdyż wymaga zbyt wielu środków różnego rodzaju (np. czasu, pamięci, materiałów); 5) a. jest zawsze ogólny, a więc może być wykorzystywany wielokrotnie do sytuacji początkowych o różnych parametrach; 6) wykonanie a. jest zawsze skończone. Wyjątkiem może być tu a. sterowania procesem ciągłym, który ma charakter nieskończony. W tej sytuacji można jednak mówić o pożądanym stanie równowagi i skończonym procesie osiągnięcia go; 7) a. jest przedmiotem abstrakcyjnym i może być zapisywany na wiele sposobów, wyrażony w różnych językach. Jest więc czymś innym, niż jego zapis. Nie pokrywa się również z zadaniem, które rozwiązuje.

Rozwój elektroniki, prowadzący do konstrukcji komputerów sprawił, że a. nabrały dużego znaczenia w związku z możliwością automatycznego ich wykonywania w postaci programów komputerowych. Wokół a. skoncentrowana jest znaczna część badań z dziedziny informatyki. Rozważa się m.in. następujące zagadnienia: a) poprawność a. — zwł. dowodzenie, że dany a. gwarantuje rozwiązanie postawionego problemu; b) szacowanie efektywności a.; c) znajdowanie najlepszych a., rozwiązujących typowe problemy; d) niezawodne przechodzenie od opisu problemu do a. rozwiązującego problem.

Na osobne omówienie zasługuje kwestia efektywności a., określana inaczej jako złożoność obliczeniowa, tzn. ilość czasu (elementarnych operacji) i miejsca w pamięci potrzebnych do realizacji a., oszacowana w odniesieniu do wielkości danych wejściowych. Klasy a. o podobnej złożoności obliczeniowej określa się zazwyczaj przy pomocy maszyny Turinga. Najważniejszymi z tych klas są a. wykonalne w czasie wielomianowym na zwykłej maszynie Turinga (P-Time) oraz na niedeterministycznej maszynie Turinga (NP-Time). Intuicyjnie można te dwie klasy rozumieć odpowiednio jako zawierające a., dla których znany jest przepis, pozwalający na znalezienie żądanej odpowiedzi w czasie określonym jako wielomian parametru danych wejściowych i takie, dla których istnieje droga do rezultatu zajmująca taki czas, choć niekoniecznie jesteśmy w stanie ją wskazać. Tę pierwszą klasę interpretuje się czasem jako klasę a. praktycznie

realizowalnych, druga natomiast zawiera większość pozostałych a., rozwiązujących ważne z praktycznego punktu widzenia problemy. Mimo intensywnych badań nie udało się dotychczas ustalić, czy klasy te się pokrywają. W przypadku, gdy praktycznie wykonalnego a. nie daje się znaleźć, stosuje się inne metody, które nie gwarantują uzyskania rezultatu, ale dają praktyczne szanse znalezienia (za pomocą symulacji komputerowej) rozwiązania przybliżonego, np. a. ewolucyjne (genetyczne), sztuczne sieci neuronowe.

**Bibliografia:** D. E. Knuth, *The Art of Computer Programming*, I–III, Reading (Mass.) 1968–1973, 1997–1998<sup>3</sup>; A. V. Aho, J. E. Hopcroft, J. D. Ullman, *The Design and Analysis of Computer A.*, Reading (Mass.) 1974 (*Projektowanie i analiza a. komputerowych*, Wwa 1983); W. M. Turski, *Propedeutyka informatyki*, Wwa 1977<sup>2</sup>, 1989<sup>3</sup>; A. V. Aho, J. E. Hopcroft, J. D. Ullman, *Data Structures and A.*, Reading (Mass.) 1983; D. Harel, *Algorithms. The Spirit of Computing*, Reading (Mass.) 1987, 1992<sup>2</sup> (*Rzecz o istocie informatyki. Algorytmika*, Wwa 1992); P. Dembiński, LFor (passim).

*Piotr Kulicki*