2. Najważniejsze własności algorytmów

Analiza algorytmów jest złożonym zagadnieniem, dlatego w podręczniku omówimy tylko jej podstawowe zasady. Dociekliwych odsyłamy do dodatkowej literatury i zachęcamy do kontynuowania zdobywania wiedzy na ten temat w przyszłości na studiach informatycznych.

Aby ułatwić zrozumienie własności algorytmów, niektóre z nich omówimy, analizując przykładowe programy napisane w językach programowania Pascal i C++.

Tworząc algorytmy, bierzemy zazwyczaj pod uwagę ich przyszłe zastosowanie w programach. Jak pamiętamy, program komputerowy jest realizacją wybranego algorytmu (wybranych algorytmów). Algorytm powinien być zatem zapisany na tyle precyzyjnie i szczegółowo, by łatwe było zrozumienie jego działania oraz zapisanie w postaci programu, przy zastosowaniu konkretnego języka programowania.

Pisząc program, chcielibyśmy, aby spełniał on kilka kryteriów:

1. rozwiązywał problem, dla którego został utworzony;
2. rozwiązywał problem dla wszystkich danych określonych w specyfikacji i odpowiednio reagował na wprowadzanie niepoprawnych danych;
3. rozwiązywał problem w sposób jak najbardziej efektywny, optymalnie wykorzystując zasoby komputera.

Najważniejsze własności algorytmów:

- poprawność;
- skończoność;
- złożoność;
- efektywność.

3. Poprawność algorytmów

Algorytm jest poprawny, jeżeli dla poprawnych danych daje poprawne wyniki. Dokładniej – dla dowolnej kombinacji danych wejściowych spełniających warunki początkowe algorytm wyprowadzi wyniki spełniające warunki końcowe.

Przypomnijmy: specyfikacja problemu (zadania) zawiera nie tylko opis danych i wyników, ale również opis związków, które między nimi zachodzą.

Algorytm jest **poprawny**, jeżeli rozwiązuje problem zgodnie ze **specyfikacją problemu (zadania)**.

Algorytm uniwersalny

to algorytm, który umożliwia rozwiązanie dowolnego zadania z pewnej klasy zadań. Przykładem może być algorytm sortowania, który umożliwia uporządkowanie dowolnego ciągu danych.

Algorytm jest **całkowicie poprawny**, jeśli dla wszystkich danych wejściowych spełniających warunki początkowe wyprowadzi wyniki spełniające warunki końcowe i obliczenia zostaną zakończone. Algorytm jest **częściowo poprawny**, jeśli dla obliczeń, które się skończą, ich wyniki są poprawne względem warunków początkowych i końcowych. W tym przypadku nie jest konieczne wykazanie, że obliczenia kończą się dla wszystkich poprawnych danych (może ich być nieskończenie wiele).

Ponadto poprawny algorytm powinien być **dobrze określony** i **uniwersalny**. Dobrze określony algorytm jest opisany w taki sposób, że występujące w nim polecenia i operacje