

Cechy algorytmów, klasyfikacja i implementacja algorytmów.

Cechy algorytmów informatycznych

Algorytmy informatyczne charakteryzuje:

- **jednoznaczność (określoność)** - wykonanie wszystkich procesów prowadzi do uzyskania poprawnego wyniku końcowego; dobry algorytm zawiera zabezpieczenia, nie dopuszczające do jego przerwania w trakcie realizacji z powodu zaistnienia nieprzewidzianych okoliczności; wiąże się ze skończonością liczby operacji
- **uniwersalność (masowość)** – algorytm ma rozwiązywać wszystkie problemy danego typu
- **szczegółowość** – polega na tym aby tak rozpisać czynności (na czynności elementarne), aby mogły być odczytane i wykonane przez realizatora algorytmu (czyli użytkownika)
- **sekwencyjność** - dotyczy kolejności wykonania operacji
- **złożoność (czasowa i pamięciowa)** - zasoby niezbędne do wykonania czynności

Jak rozwiązywać algorytmy?

Czynności, które służą do rozwiązania zadania to **kroki**. Używamy ich w określonej kolejności:

1. analiza treści zadania
2. wykaz danych wejściowych; wiadomych i niewiadomych oraz relacji między nimi
3. sprawdzenie czy zadanie posiada jednoznaczne rozwiązanie
4. wybór metody rozwiązania zadania
5. opis czynności, które należy wykonać z danymi wejściowymi przy zastosowaniu wybranej metody rozwiązania
6. sporządzenie i przedstawienie wyników rozwiązania zadania

Budowa algorytmu

Wykonawcą algorytmu może być komputer lub człowiek.

Do podstawowych elementów potrzebnych do napisania algorytmu należą:

- nazwa algorytmu
- opis obiektów
- deklaracja stałych i zmiennych tekstowych oraz liczbowych
- deklaracja funkcji użytkownika
- opis czynności jakie należy wykonać z obiektami, co realizujemy za pomocą **instrukcji**, które opisują nie tylko sposób działania i kolejność ich wykonywania ale również ewentualne warunki jakie muszą być spełnione w celu uzyskania prawidłowego rozwiązania
- opis wyników, który zawiera sposób udostępnienia wyników rozwiązanego zadania

Implementacja algorytmu

Język programowania pozwala programiście na precyzyjne przekazanie maszynie, jakie dane mają ulec obróbce i jakie czynności należy podjąć w określonych warunkach.

Język programowania pozwala na **precyzyjny zapis algorytmów oraz innych zadań, jakie komputer ma wykonać (implementacja)**.

Najbardziej pożądane cechy języka programowania:

- łatwość formułowania kodu - zadań dla procesora
- możliwie najszersze wykorzystanie możliwości procesora poprzez składnię języka



Klasyfikacja algorytmów

Jest dość ciężko dokonać jednolitej klasyfikacji algorytmów, dlatego zostaną przedstawione podstawowe **paradygmaty** czyli wzorce tworzenia algorytmów komputerowych:



- **dziel i zwyciężaj** – dzielimy problem na kilka mniejszych, a te znowu dzielimy, aż ich rozwiązania staną się oczywiste
- **programowanie dynamiczne** – problem dzielony jest na kilka, ważność każdego z nich jest oceniana i po pewnym wnioskowaniu wyniki analizy niektórych prostszych zagadnień wykorzystuje się do rozwiązania głównego problemu
- **metoda zachłanna** – nie analizujemy podproblemów dokładnie, tylko wybieramy najbardziej obiecującą w tym momencie drogę rozwiązania



- **programowanie liniowe** – oceniamy rozwiązanie problemu przez pewną funkcję jakości i szukamy jej minimum
- **poszukiwanie i wyliczanie** – kiedy przeszukujemy zbiór danych aż do odnalezienia rozwiązania
- **algorytm probabilistyczny** – algorytm działa poprawnie z bardzo wysokim prawdopodobieństwem, ale wynik nie jest pewny
- **heurystyka** – człowiek na podstawie swojego doświadczenia tworzy algorytm, który działa w najbardziej prawdopodobnych warunkach, rozwiązanie zawsze jest przybliżone

Proces implementacji

Każdy algorytm komputerowy musi być wprowadzony do komputera w bardzo rygorystycznie zdefiniowanym języku. Ludzie często komunikując się przesyłają między sobą informacje wieloznaczne. Komputery mogą reagować tylko na całkowicie jednoznaczne instrukcje. Jeżeli dany algorytm da się wykonać na maszynie o dostępnej **mocy obliczeniowej** i **pamięci** oraz akceptowalnym czasie, to mówi się, że jest **obliczalny**.



Poprawne działanie większości **algorytmów implementowanych** w komputerach opiera się na kolejnej realizacji pewnego **zestawu warunków**. Jeżeli, któryś z nich nie zostanie spełniony, to program kończy się komunikatem błędu. Czasami podczas implementacji algorytmu ważny warunek zostanie pominięty. Dla przykładu, mamy program dzielący przez siebie dwie liczby. Użytkownik poleca wykonać dzielenie przez zero. Działanie aplikacji, która nie sprawdzi warunku „dzielnik nierówny zero”, zostanie przerwane przez system operacyjny komputera.

Zadanie

Dokonaj implementacji w wybranym języku programowania (np. C++) poniższych algorytmów:

