

# **Laboratorium Informatyki I**

## **Ćwiczenie 1**

### **ALGORYTMIKA**

#### **1. Wprowadzenie**

Programowanie w stylu Top-Down, zalecanym szczególnie dla początkujących programistów, wymaga wstępnego przeanalizowania problemu i jego analizy od zadań globalnych programu, poprzez coraz mniejsze do poziomu podstawowego. Realizować to można przygotowując opisowy opis poszczególnych kroków programu, uzupełniając taki opis tzw. pseudokodem. Jednakże znacznie wygodniejsze i przejrzystsze jest zapisanie algorytmu programu za pomocą schematu. Zastosowanie takiego sposobu opisu projektu programu jest także czytelniejsze dla innych osób, które będą wspomagały pisanie programu lub analizowały jego działanie.

#### **2. Algorytmy**

**Algorytm** to zestaw instrukcji lub logicznych kroków służących rozwiązaniu problemu lub wykonaniu określonego zadania. Jest więc on swoistym przepisem na realizację określonego problemu za pomocą kodu w wybranym języku programowania. Można go zapisać za pomocą:

- **Opisu słownego** – w formie „literackiej” opisać sposób realizacji danego problemu
- **Pseudo kodem** – skrócona forma opisu słownego w którym niektóre czynności „standardowe” zapisuje się za pomocą słów kluczowych zaczerpniętych z popularnych języków programowania (`for`, `if`, `def`, `function`,...).
- **Schematu graficznego** (blokowy, NS lub UML) wykorzystanie symboli graficznych do zobrazowania sposobu przetwarzania informacji w celu rozwiązania problemu

Praca nad algorytmem jest pierwszym etapem projektowania aplikacji realizującej określone zadanie. Wybór optymalnej formy zapisu algorytmu jest bardzo istotny dla efektywności prac projektowych, szczególnie gdy praca odbywa się w grupie. Zapis za pomocą schematu graficznego wydaje się najkorzystniejszy przede wszystkim ze względu na jego czytelność i łatwość wprowadzania modyfikacji.

W większości projektów można opisać jako system powiązanych ze sobą algorytmów realizujących pojedyncze zadania. Można je podzielić na dwie zasadnicze grupy: algorytmy ogólnego przeznaczenia i wyspecjalizowane. Do pierwszej grupy należą:

- Algorytm liniowy (sekwencyjny) wykonujący ciąg określonych zadań
- Algorytm warunkowy (rozgałęziony) , pozwalający na adaptacyjne wykonanie określonych czynności, zazwyczaj w zależności od spełnienia określonych warunków logicznych
- Algorytm iteracyjny (z pętlą lub cykliczny), powtarzający określona listę zadań
- Algorytm rekurencyjny, pozwalający na kaskadowe wykonywanie samego siebie

W literaturze dostępne są opisy (algorytmy) rozwiązujące często występujące przy programowaniu problemy, które można zaimplementować w opracowywanym algorytmie. Należą do nich:

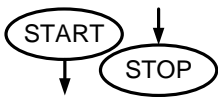
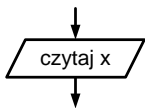
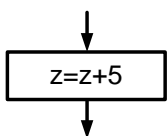
- Algorytmy wyszukiwania konkretnych elementów w zbiorze
  - Przeszukiwanie liniowe
  - Wyszukiwanie binarne
  - Algorytm Dijkstry
  - Algorytm Kruskala
- Algorytmy sterowania, porządkujące elementy w zbiorze
  - Sortowanie bąbelkowe
  - Sortowanie przez wstawienie
  - Sortowanie przez wybór
  - Sortowanie przez scalenie
- Algorytmy dzielenia i zwyciężania, podziału na mniejsze podproblemy, rozwiązywaniu ich niezależnie a następnie łączeniu w celu uzyskania wyniku końcowego
  - Algorytm mnożenia macierzy Strassena
  - Algorytm Karatsuby do mnożenia dużych liczb
  - Algorytm Cooleya-Tukeya do szybkiej transformaty Fouriera
- Algorytmy chciwe, zakłada się że optymalizacja na poziomie lokalnym (mniejszych problemów) zapewni optymalizację problemu globalnego.
  - Algorytm Kruskala do znajdowania minimalnego drzewa rozpinającego
  - Algorytm Prima do znajdowania minimalnego drzewa rozpinającego
  - Algorytm Huffmana do kodowania danych
- Algorytmy dynamiczne, rozwiązywanie złożonych problemów poprzez dzielenie ich na mniejsze podproblemy i przechowywanie rozwiązań tych podproblemów, aby uniknąć ich ponownego przeliczania
  - Algorytm Fibonacciego
  - Algorytm Dijkstry do znajdowania najkrótszej ścieżki
  - Algorytm Bellmana-Forda do znajdowania najkrótszej ścieżki

- Algorytmy probabilistyczne, wykorzystanie metod losowych i probabilistycznych do rozwiązywania problemów
  - Algorytm Monte Carlo
  - Algorytm losowego sortowania szybkiego
  - Algorytm Rabina-Karpa do wyszukiwania wzorców
- Algorytmy cofania się, analizowanie wszystkich potencjalnych rozwiązań i odrzucania niespełniających przyjętych kryteriów
  - Algorytm rozwiązujący problem N królowych
  - Algorytm rozwiązania problemu komiwojażera
  - Algorytm rozwiązujący problem plecaka
- Algorytmy genetyczne, oparte na ewolucji naturalnej, wykorzystują mechanizmy selekcji, krzyżowania i mutacji w poszukiwaniu optymalnego rozwiązania
  - Algorytm optymalizacji trasy dostawy
  - Algorytm harmonogramowania
  - Algorytm projektowania sieci telekomunikacyjnej
- Algorytmy uczenia maszynowego, oparte na systemach SI, automatycznego uczenia się w oparciu o dostępne wzorce (dane)
  - Algorytm regresji liniowej
  - Naiwny algorytm klasyfikacji Bayesa
  - Algorytm drzewa decyzyjnego
  - Sztuczny algorytm sieci neuronowej

### 3. Schematy blokowe

Podstawowym (najbardziej intuicyjnym) sposobem zapisu graficznego algorytmu są schematy blokowe. Składa on się z figur płaskich, obrazujących różne działania oraz strzałek ukazujących powiązania pomiędzy poszczególnymi elementami schematu, oraz kierunku przepływu informacji.

Tabela 1. Podstawowe elementy schematów blokowych

Nazwa	Opis	Rysunek
Blok graniczny	Obrazuje początek, koniec, przerwanie lub wstrzymanie wykonywania działania.	
Blok wejścia-wyjścia	Odpowiada procesowi wprowadzeniu danych do programu i wyprowadzeniu wyników obliczeń	
Blok operacyjny	Obrazuje wykonanie operacji, w wyniku której zmienia się wartość, postać lub miejsce zapisu danych	

Blok decyzyjno-warunkowy	Ukazuje możliwość wyboru dwóch wariantów wykonania programu po sprawdzeniu zapisanego warunku	
Blok podprogramu	Pokazuje blok oddzielnie zdefiniowanego podprogramu, np. procedury lub funkcji	
Blok komentarza	Wprowadza komentarz objaśniający wskazany fragment algorytmu	
Łącznik wewnątrz-stronicowy	Umożliwia łączenie odrębnych części schematu na tej samej stronie dokumentu	
Łącznik między-stronicowy	Umożliwia łączenie odrębnych części schematu na różnych stronach dokumentu	

## 4. Zadania

Dla podanych poniżej zadań zaproponować algorytm oraz zapisać go za pomocą schematu blokowego.

- Zaproponować algorytm liczenie pierwiastków równania kwadratowego  $ax^2+bx+c=0$
- Zaproponować algorytm obliczający dla zadanego przedziału liczb całkowitych sumę liczb parzystych oraz iloczyn liczb nieparzystych.
- Zaproponować algorytm obliczania średniej geometrycznej z szeregu liczb całkowitych podawanych przez użytkownika do czasu wprowadzenia liczby 0
- Opracować algorytm obliczający pole trójkąta na podstawie podanych przez użytkownika długości boków. (algorytm musi sprawdzać czy z podanych odcinków można zbudować trójkąt)
- Zaproponować algorytm, który wyświetla wprowadzone przez użytkownika trzy liczby w kolejności rosnącej
- Opracować algorytm liczący silnię z podanej przez użytkownika liczby