



Wybrane elementy informatyki - wykład 3

Algorytmy

dr Marcin Ziótkowski

Instytut Matematyki i Informatyki
Akademia im. Jana Długosza w Częstochowie

16 marca 2017 r.

ALGORYTM

Algorytm - to sposób rozwiązania danego zagadnienia (problemu). W informatyce oraz innych naukach ścisłych wymagamy, żeby sposób ten rozwiązywał dany problem w skończonej ilości kroków i w skończonym czasie i używał skończonej ilości zasobów pamięci. Zatem algorytm to najczęściej lista operacji, które prowadzą do rozwiązania danego problemu. Lista jest tutaj rozumiana jako ciąg, więc w algorytmie ważne są nie tylko operacje prowadzące do rozwiązania tego problemu, ale także ich kolejność.

Pierwsze poważne algorytmy były tworzone w celu rozwiązywania problemów matematycznych np. algorytm znajdowania przybliżeń pierwiastków kwadratowych czy algorytm Euklidesa znajdowania największego wspólnego dzielnika dwóch dodatnich liczb naturalnych.

Są różne sposoby zapisu algorytmów:

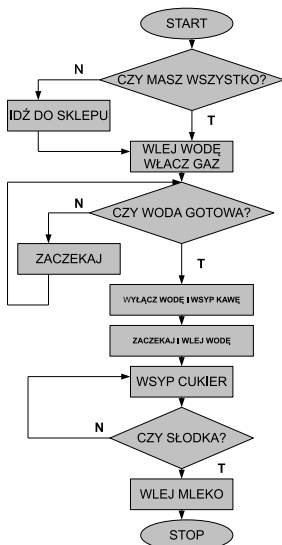
- 1 Lista kroków - podajemy ciąg operacji, które prowadzą od danych wejściowych do żądanego rezultatu
- 2 Pseudokod - ogólny nieistniejący sztuczny język programowania, w którym zapisujemy algorytm
- 3 **Schematy blokowe** - układ różnych figur geometrycznych oraz strzałek łączących te figury, które łącznie pokazują operacje oraz ich kolejność - najbardziej uniwersalny sposób zapisu algorytmów

Algorytm jest pierwszym krokiem przy rozwiązywaniu różnych zagadnień informatycznych. Na jego podstawie tworzy się potem program w wybranym języku programowania.

PRZYKŁADY ALGORYTMÓW - lista kroków - ALGORYTM PRZYGOTOWYWANIA KAWY

- 1 Wejdź do kuchni i sprawdź czy masz kawę, cukier i mleko. Jeżeli tak - idź do punktu 3, jeśli nie - idź do punktu 2
- 2 Idź do sklepu, kup brakujące produkty i idź do punktu 3
- 3 Wlej do czajnika wodę i włącz gaz
- 4 Sprawdź czy woda się zagotowała, jeżeli tak - idź do punktu 6, jeżeli nie - idź do punktu 5
- 5 Zaczekaj chwilę i idź do punktu 4
- 6 Wyłącz wodę i wsyp kawę do kubka
- 7 Poczekaj chwileczkę i wlej wodę do kubka
- 8 Przykryj kubek i poczekaj chwileczkę
- 9 Wsyp odrobinę cukru do kawy
- 10 Spróbuj kawę, jeżeli jest wystarczająco słodka idź do punktu 11, jeżeli nie idź do punktu 9
- 11 Wlej odrobine mleka i zamieszaj

ALGORYTM PRZYGOTOWYWANIA KAWY- schemat blokowy



PODSTAWOWE RODZAJE ALGORYTMÓW

- 1 ALGORYTM SEKWENCYJNY - wykonujemy po kolei wszystkie operacje bezwarunkowo, w takim algorytmie nie może być operacji podjęcia decyzji oraz powtarzania operacji
- 2 ALGORYTM ROZGAŁĘZIONY - algorytm, w którym w zależności od spełnienia pewnego warunku wykonujemy różne operacje, algorytm taki zawiera przynajmniej jedną operację podjęcia decyzji (instrukcję warunkową)
- 3 ALGORYTM ITERACYJNY - algorytm, w którym występuje wielokrotne powtarzanie tych samych operacji, w algorytmie takim występują tzw. pętle czyli ciągi operacji, które są wykonywane wielokrotnie

Algorytmy tworzymy po to, aby "nauczyć" komputer rozwiązywać dany problem. Przekazujemy maszynie nasz tok myślenia, aby potem ją wykorzystać w celu szybkiego wykonywania żmudnych obliczeń w oparciu o duże szybkości procesorów i duże zasoby pamięci.

ZASADY TWORZENIA SCHEMATÓW BLOKOWYCH

W schematach blokowych używamy figur geometrycznych oraz połączeń między nimi. Każdy schemat zaczyna się od jednego klocka startowego i kończy się przynajmniej jednym klockiem końcowym.

Oto podstawowe klocki używane przy tworzeniu algorytmów i operacje, które mogą być wpisane w dane klocki:

- 1 ELIPSY - są zarezerwowane dla klocków startowych i końcowych - START ORAZ STOP
- 2 PROSTOKĄTY - są zarezerwowane na pojedyncze lub wielokrotne operacje wykonywane na danych, w prostokątach są zapisywane operacje podstawienia oraz podstawowych działań arytmetycznych
- 3 RÓWNOLEGŁOBOKI - są używane do wprowadzania danych oraz wyprowadzania wyników częściowych i końcowych
- 4 ROMBY - są używane dla bloków decyzyjnych - instrukcji wyboru, w algorytmie muszą być zawsze rozważane obie drogi zależne od dokonanego wyboru

Zauważmy również że bloki decyzyjne będą miały również zastosowanie w przypadku realizacji tzw. pętli czyli ciągów instrukcji wykonywanych wielokrotnie w zależności od spełnienia pewnego warunku nazywanego WARUNKIEM KONTYNUACJI PĘTLI.

Przedstawimy teraz wiele przykładów schematów blokowych.

PRZYKŁAD 1

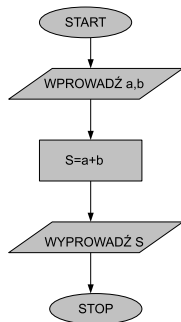
Przedstawmy algorytm dodawania dwóch liczb naturalnych.

Specyfikacja problemu:

dane wejściowe: dwie liczby naturalne a , b

dane wyjściowe, rezultat: suma liczb a oraz b

Oto schemat blokowy realizujący dane zagadnienie.



PRZYKŁAD 2

Przedstawmy algorytm dzielenia dwóch liczb rzeczywistych.

Specyfikacja problemu:

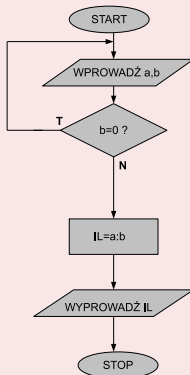
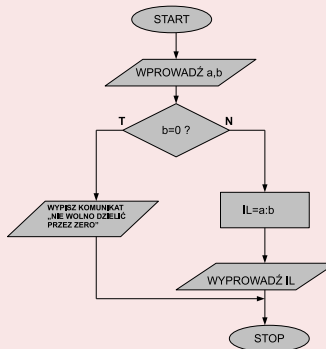
dane wejściowe: dwie liczby rzeczywiste a , b

dane wyjściowe, rezultat: iloraz liczb a oraz b

Ponieważ mamy wykonać dzielenie, druga z liczb nie może być równa zero.

Oto schematy blokowe realizujące dane zagadnienie. Drugi z algorytmów jest już algorytmem zawierającym pętlę.

ALGORYTM DZIELENIA



PRZYKŁAD 3

Przedstawmy algorytm rozwiązywania równania liniowego
 $ax + b = 0$.

Specyfikacja problemu:

dane wejściowe: dwie liczby rzeczywiste a , b - współczynniki równania

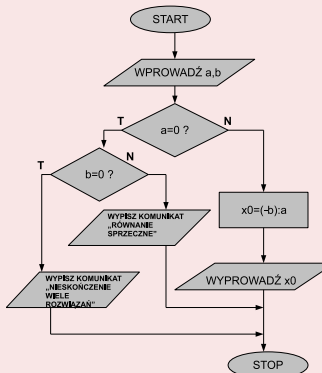
dane wyjściowe, rezultat: rozwiązanie równania

Przypomnijmy, że są możliwe trzy sytuacje:

- Gdy $a \neq 0$ - wówczas równanie ma jedno rozwiązanie $x_0 = -\frac{b}{a}$
- Gdy $a = 0$ oraz $b = 0$ - równanie ma nieskończenie wiele rozwiązań
- Gdy $a = 0$ oraz $b \neq 0$ - równanie jest sprzeczne

Wystarczy wykorzystać dwa bloki decyzyjne, aby zbudować algorytm rozwiązujący ten problem.

ALGORYTM ROZWIĄZYWANIA RÓWNIANIA LINIOWEGO



Prawdziwą potęgą jest stosowanie wielokrotnych powtórzeń tych samych ciągów operacji czyli stosowanie tzw. PĘTLI ITERACYJNYCH. Z użyciem pętli można rozwiązywać nawet bardzo skomplikowane problemy.

W algorytmach mamy do czynienia z dwoma rodzajami pętli:

- 1 Pętle, dla których wiadoma jest liczba powtórzeń ciągu operacji - w takich pętlach wprowadzamy licznik, który zlicza ilość powtórzeń pętli
- 2 Pętle, dla których liczba powtórzeń ciągu operacji jest nieznana - takie pętle kończą się w sytuacji niespełnienia danego warunku kontynuacji pętli niezależnego od liczby powtórzeń

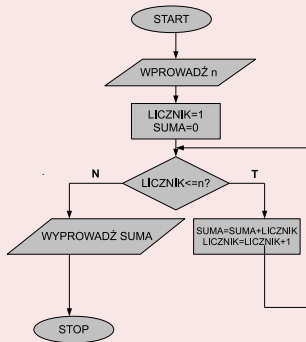
PRZYKŁADY ALGORYTMÓW ITERACYJNYCH

ALGORYTM OBLICZANIA SUMY $1 + 2 + 3 + 4 + \dots + n$

dane wejściowe: n - liczba naturalna dodatnia wprowadzona przez użytkownika

dane wyjściowe: suma $1 + 2 + 3 + \dots + n$

liczba powtórzeń pętli: n

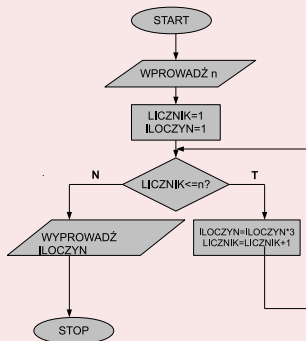


ALGORYTM OBLICZANIA POTĘGI 3^n

dane wejściowe: n - liczba naturalna wprowadzona przez użytkownika

dane wyjściowe: potęga 3^n

liczba powtórzeń pętli: n

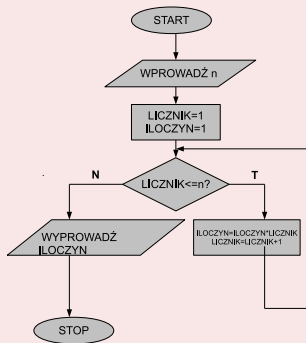


ALGORYTM OBLICZANIA SILNI

dane wejściowe: n - liczba naturalna wprowadzona przez użytkownika

dane wyjściowe: liczba $n! = 1 \cdot 2 \cdot 3 \cdot \dots \cdot n$

liczba powtórzeń pętli: n

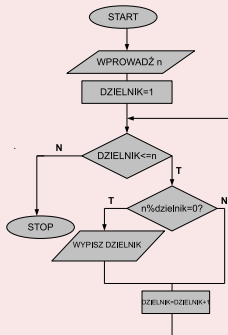


ALGORYTM WYPISYWANIA DZIELNIKÓW

dane wejściowe: n - liczba naturalna dodatnia wprowadzona przez użytkownika

dane wyjściowe: dzielniki liczby n

liczba powtórzeń pętli: n

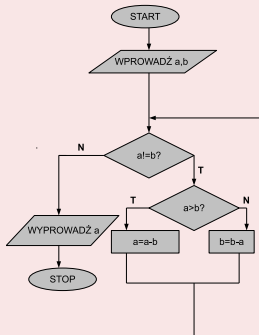


ALGORYTM EUKLIDESA OBLICZANIA NWD

dane wejściowe: a, b - liczby naturalne dodatnie wprowadzone przez użytkownika

dane wyjściowe: $\text{NWD}(a,b)$ (największy wspólny dzielnik liczb a,b)

liczba powtórzeń pętli: nieznana

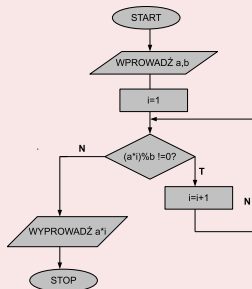


ALGORYTM OBLICZANIA NWW

dane wejściowe: a, b - liczby naturalne dodatnie wprowadzone przez użytkownika

dane wyjściowe: $NWW(a,b)$ (najmniejsza wspólna wielokrotność liczb a,b)

liczba powtórzeń pętli: nieznana



PRZYKŁADY ALGORYTMÓW ITERACYJNYCH

ALGORYTM SPRAWDZANIA CZY LICZBA JEST PIERWSZA

dane wejściowe: n - liczba naturalna (większa od 1) wprowadzona przez użytkownika

dane wyjściowe: informacja o pierwszości liczby

liczba powtórzeń pętli: nieznaną

