

Algorytmy i struktury danych

Definicja i cechy algorytmów
Sieci działań
Programowanie strukturalne

Witold Marańda
maranda@dmcs.p.lodz.pl

1

Literatura

- 1. Niklaus Wirth,**
Algorytmy + Struktury danych = Programy,
WNT, Warszawa 1999
- 2. David Harel,**
Rzecz o istocie informatyki - Algorytmika,
WNT, Warszawa 2000

2

Algorytm

Przepis na rozwiązanie określonego problemu za pomocą prostych czynności wykonywanych w ściśle określonej kolejności.

Czynności:

1. muszą być znacznie prostsze od realizowanego algorytmu
2. muszą być wykonywalne dla danego „sprzętu”
(prostota czynności jest sprawą względną)

Kolejność:

1. określenie kolejności wykonywania czynności jest krytyczne dla osiągnięcia celu algorytmu;
2. musi istnieć mechanizm rozgałęziania algorytmu, tj. decydowania o kolejności w trakcie wykonywania algorytmu na podstawie zaistniałych warunków.

3

Cechy algorytmu

Skończoność:

algorytm musi zapewnić osiągnięcie do rozwiązania w skończonej liczbie kroków (a więc też w skończonym czasie).

Skończona liczba kroków nie oznacza, że z góry wiadomo po ilu krokach algorytm się zakończy.

Komunikat o błędzie lub braku rozwiązania też jest jednym z możliwych poprawnych zakończeń realizacji algorytmu.

np.

Obliczanie wartości $\sin(x)$ można wykonać numerycznie za pomocą sumowania kolejnych wyrazów szeregu:

$$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \frac{x^9}{9!} - \dots$$

Algorytm taki musi posiadać warunek zakończenia tej operacji (np. kryterium dokładności) aby nie wykonywał się, mimo że poprawnie, w nieskończoność.

4

Cechy algorytmu

Kompletność:

algorytm musi uwzględniać wszystkie możliwe przypadki, które mogą pojawić się podczas jego realizacji.

Uwzględnienie różnych przypadków oznacza zapewnienie dalszej realizacji algorytmu, zgodnie z przewidzianymi na taką okoliczność instrukcjami.

W praktyce programistycznej oznacza to przewidzenie wystąpienia błędów numerycznych i logicznych oraz opracowanie systemu reakcji (komunikaty o błędach, odpowiednie zakończenie działania).

np.

Obliczanie rozwiązań równania kwadratowego wymaga uwzględnienia przypadków:

$$b^2 - 4ac > 0, \quad b^2 - 4ac < 0$$

Brak sprawdzenia trzeciego warunku ($=0$) i wartości parametru a jest przypadkiem niekompletności algorytmu i może spowodować jego błędne działanie.

5

Cechy algorytmu

Jednoznaczność:

dla tych samych danych wejściowych algorytm musi zawsze dawać te same wyniki.

Jednoznaczność w praktyce oznacza niezależność działania programu od momentu jego wykonania, wpływu innych programów realizowanych równocześnie przez system operacyjny oraz, co najtrudniejsze, od sprzętu realizującego dany algorytm.

np.

Algorytmy wykonujące obliczenia arytmetyczne powinny dawać dokładnie takie same wyniki na różnych komputerach i systemach operacyjnych - jest to bardzo trudne do spełnienia (różne kodowanie liczb, różne algorytmy ich przetwarzania)

Algorytmy formatujące tekst (procesory tekstu) powinny dawać taki sam wygląd strony (układ tekstu, łamanie wyrazów, etc.) zgodny z informacją zapisaną w pliku, niezależnie od typu komputera i wersji systemu operacyjnego ☺

6

Terminologia

- Algorytm może być wykonany przez każdego (człowieka lub maszynę), kto zna język, w którym zapisano algorytm.
- Wykonawca algorytmu musi umieć wykonywać jego instrukcje, ale nie musi znać jego istoty algorytmu lub jego przeznaczenia, aby osiągnąć pożądaną wynik.
- Języki przeznaczone do zapisu algorytmów w postaci instrukcji zrozumiałych przez maszynę cyfrową noszą nazwę „języków programowania”.
- Algorytm zapisany w języku programowania nosi nazwę: „programu”.
- Języki programowania (wysokiego poziomu) są kompromisem pomiędzy językiem naturalnym (zrozumiałym dla człowieka) a pojęciami bliskimi konstrukcji maszyny cyfrowej (bity, bajty, etc.)

7

Dane algorytmu

Każdy algorytm wykonuje operacje na obiektach: liczbach i ich bardziej złożonych strukturach.

*liczby, wektory, tablice, rekordy, struktury, unie, stosy
kolejki, listy, drzewa, grafy ...*

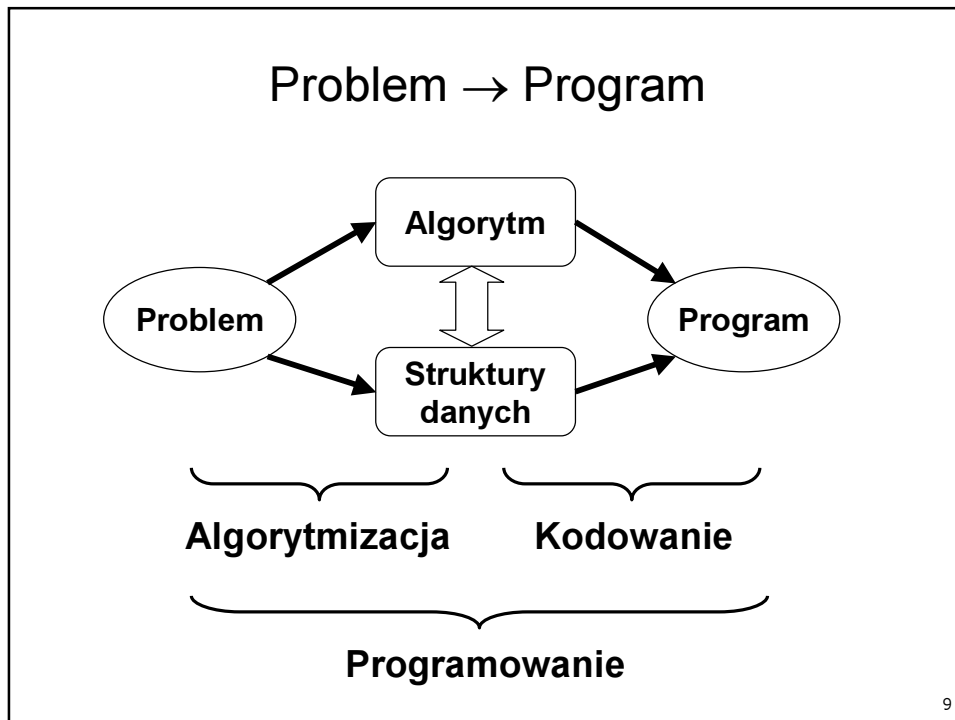
Liczby są zwykle modelami obiektów rzeczywistych, w aspekcie tych cech, którymi zajmuje się algorytm.

- *wartość pojedynczej liczby może być modelem wysokości zarobku w programie finansowym,*
- *wektor trzech liczb może być modelem punktu w przestrzeni w programie grafiki trójwymiarowej,*
- *kolejka może być modelem zgłoszeń zapytań w programie bazy danych, etc.*

Konstrukcja każdego algorytmu „stoi na dwóch nogach”:

- algorytmizacji problemu (rozbiciu na elementarne operacje),
- doborze właściwych struktur danych, stosownych do zadania, którego dotyczy algorytm.

8



Zapis algorytmu

Zwięzłość, czytelność i wysoki poziom abstrakcji.

Sieć działań (schemat blokowy):

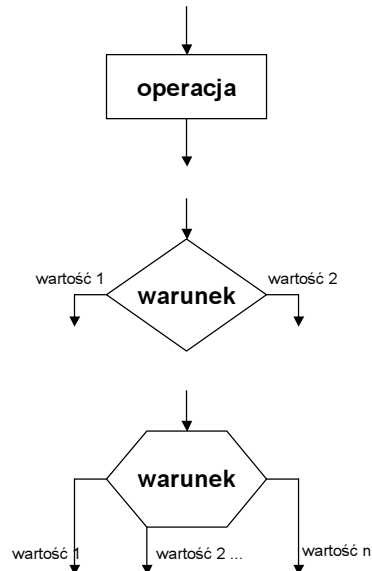
1. Elementarne czynności oznaczone są blokami (węzły sieci), a kolejność wyznaczona jest poprzez gałęzie sieci, łączące węzły.
2. Kształt bloków odpowiada rodzajowi operacji, a strzałki gałęzi identyfikują jednoznacznie ich kolejność.
3. Niezależność struktury algorytmu od architektury konkretnej maszyny i rodzaju kodowania liczb.

Za pomocą sieci działań możliwe jest zapisanie każdego poprawnego algorytmu!

Każdy algorytm można zapisać za pomocą wielu różnych sieci działań!

10

Sieć działań



blok operacyjny (obliczeniowy)
dokładnie jedno wejście i jedno wyjście

realizacja elementarnej czynności

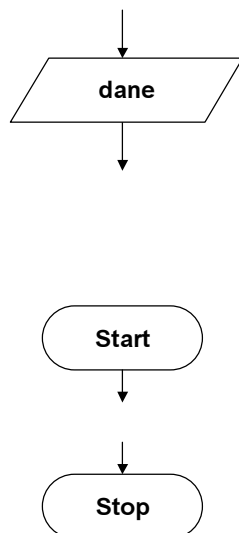
blok decyzyjny (warunkowy)
dokładnie jedno wejście, dwa
lub więcej wyjść

warunek jest zdaniem (wyrażeniem
arytmetycznym), które w danej sytuacji
może przyjmować różne wartości

blok warunkowy jest miejscem rozgałęziania
programu: kolejna instrukcja do wykonania
jest wyznaczana przez gałąź z opisem wartości
równym wartości testowanego wyrażenia

11

Sieć działań



blok wejścia-wyjścia (we/wy)
dokładnie jedno wejście i jedno
wyjście

oznaczenia miejsca wymiany danych
pomiędzy algorytmem a światem
zewnętrznym, bez konkretyzowania
źródła danych i metody tej operacji

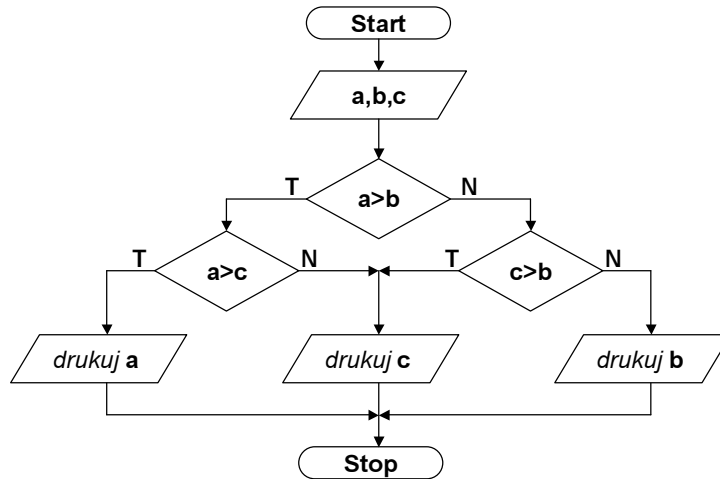
blok startu
dokładnie jedno wyjście
oznaczenia początku algorytmu

blok stopu
dokładnie jedno wejście
oznaczenia końca algorytmu

12

Przykład A

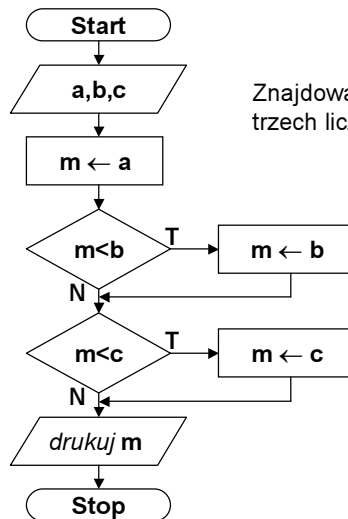
Znajdowanie największej z trzech liczb: $\max(a,b,c)$



13

Przykład B

Znajdowanie największej z trzech liczb: $\max(a,b,c)$



14

Sieci A vs B ?

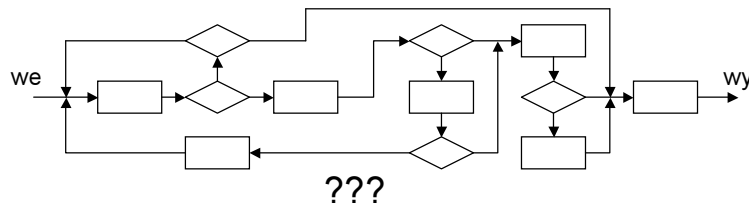
A	B
max(a,b,c)	max(a,b,c)
rozmiar:	rozmiar:
4 we/wy	2 we/wy
3 dec.	2 dec.
0 op.	3 op.
szybkość:	szybkość:
👍 2 we/wy	2 we/wy
2 dec.	2 dec.
0 op.	1..3 op.
rozbudowa:	rozbudowa:
trudna	łatwa 👍

Rozwiązanie problemu możliwe jest za pomocą różnych sieci działań.

Optymalizacja algorytmu polega na znalezieniu sieci najlepszej według określonego kryterium: np. rozmiaru, szybkości, rozbudowy.

15

Analiza algorytmów



Analiza poprawności sieci działań jest zadaniem skomplikowanym. Prześledzenie wszystkich ścieżek przejścia przez sieć dla wszystkich możliwych danych jest zwykle niemożliwe.

W praktyce testowanie algorytmów dotyczy sytuacji:

- typowych
- krytycznych

Techniki konstrukcji sieci wspomagające analizę poprawności to:

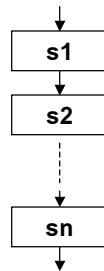
- podział na moduły (podprogramy)
- tworzenie sieci strukturalnych (programowanie strukturalne)

16

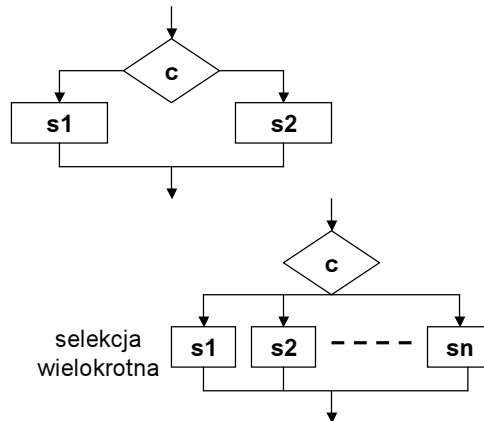
Sieci strukturalne

Podstawowe rodzaje konstrukcji strukturalnych:

sekwencja



selekcja

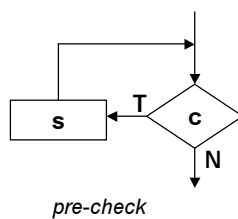


17

Sieci strukturalne

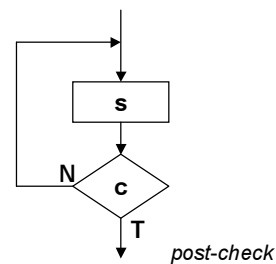
Podstawowe rodzaje konstrukcji strukturalnych:

cykl (pętla)



pre-check

Instrukcja s będzie wykonana 0 lub więcej razy. Cykl jest powtarzany dopóki spełniony jest warunek c.



post-check

Instrukcja s będzie wykonana 1 lub więcej razy. Cykl jest powtarzany aż do spełnienia warunku c.

18

Algorytmy strukturalne

Każda konstrukcja strukturalna (sekwencja, selekcja, cykl) posiada dokładnie jedno wejście i jedno wyjście.

Sekwencja

składa się z instrukcji, która są wykonywane w danej konstrukcji tylko jedno raz w ściśle określonej kolejności.

Selekcja

musi uwzględniać wszystkie możliwe wartości, które może przyjmować warunek selekcji (kompletność).

Cykl

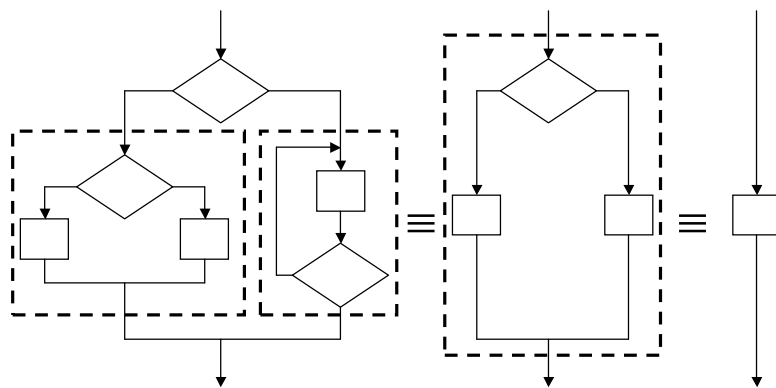
nie może być nieskończony (skończoność), a zmienna występująca w warunku (zmienna sterująca cyklu) musi być modyfikowana wewnątrz cyklu.

Za pomocą sekwencji, selekcji i cykli można zapisać każdy algorytm.

Algorytmy zapisane jedynie za pomocą sekwencji, selekcji i cykli nazywają się algorytmami strukturalnymi (sieci strukturalne)

19

Struktura modułowa



Algorytmy strukturalne można redukować, zastępując złożone fragmenty blokami o wyższym stopniu funkcjonalności.

Podział na moduły ułatwia testowanie całego programu, gdyż można zapewnić oddzielnie poprawność modułów (*black box*), a następnie całej, zredukowanej sieci.

20

Programowanie strukturalne

Konstrukcja strukturalne (sekwencje, selekcje, cykle) są realizowane w językach programowania jako tzw. instrukcje sterujące.

Programowanie strukturalne polega na konstruowaniu programów wykorzystując jedynie struktur sekwencji, selekcji i cykli.

Zalety programowania strukturalnego:

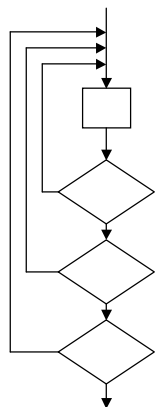
- hierarchiczna i modułowa struktura programu
- możliwość tworzenia programu w stylu *top-down* lub *bottom-up*
- możliwość analizy poprawności programu
- łatwość modyfikacji, redukcja błędów i czasu pisania programu
- możliwość wykorzystania modułów do innych programów

21

Programowanie strukturalne

Języki strukturalne

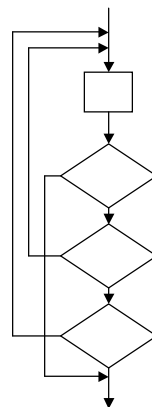
Pascal, C, Java ...



siec strukturalna

Języki niestukturalne

Fortran, Basic ... (*goto*)



siec niestukturalna
(*spaghetti programming*)

22