

Technologia informacyjna

Algorytm

Janusz Uriasz

Algorytm

Algorytm - (łac. algorithmus); ścisły przepis realizacji działań w określonym porządku, system operacji, reguła komponowania operacji, sposób postępowania. Algorytmy wyznaczają w postaci kolejnych faz (etapów) tok (sposób) postępowania w określonych sytuacjach. Są one często odkrywane i formułowane przez teoretyków i wchodzą w skład ogólnej wiedzy o sprawnym i skutecznym działaniu. Ich formułowanie umożliwia rozwiązywanie podobnych zadań należących do określonej klasy. Schematy algorytmiczne wiążą się z językiem programowania.

Właściwości algorytmu:

- Poprawność – dla każdego zestawu poprawnych danych wejściowych algorytm powinien dawać poprawne wyniki
- Skończoność – dla każdego zestawu poprawnych danych wejściowych algorytm powinien dać wyniki w skończonej liczbie operacji
- Określoność – algorytm nie może pozostawiać wątpliwości co do wyboru kolejnych operacji
- Efektywność (sprawność)-algorytm rozwiązuje zadanie w „rozsądnym” czasie (najniższy koszt, najmniejsza liczba kroków)
- Ogólność (uniwersalność) – algorytm powinien rozwiązywać nie jedno szczególne zadanie lecz całą klasę zadań
- Realizator algorytmu potrafi wykonać wszystkie operacje 😊

Analiza algorytmów – szukanie najlepszych algorytmów dla programów komputerowych

Odpowiedzi na pytania:

1. Czy dany problem może być rozwiązany z wykorzystaniem komputera w dostępnym czasie i przy dostępnej pamięci?
 2. Który ze znanych algorytmów należy zastosować w danych okolicznościach?
 3. Czy istnieje lepszy algorytm od rozważanego?
 4. Czy to algorytm optymalny?
 5. Jak uzasadnić, że stosując dany algorytm, rozwiąże się postawione zadanie?
-

Analiza algorytmów

1. Poprawność semantyczna (program wykonuje postawione – sformułowane- zadanie)
 2. Prostota
 3. Czas działania
 4. Ilość zajmowanej pamięci
 5. Optymalność
 6. Okoliczności, w jakich należy go używać
-

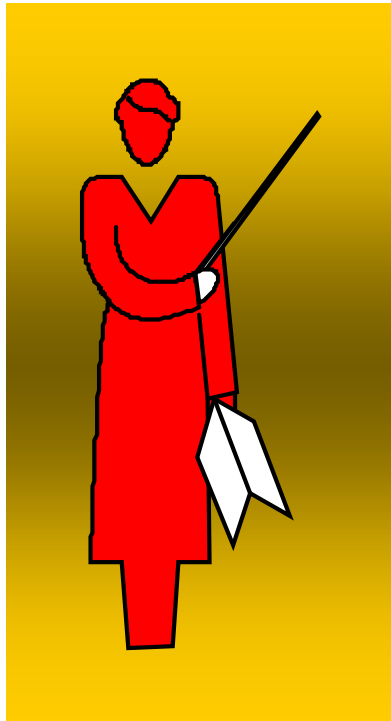
Prezentacja algorytmu (sposób zapisu):

Opisowa – lista kroków

Zadanie nawigacyjne:

1. Dane współrzędne punktu $P(B,L)$, linia geodezyjna s , Azymut A
2. Oblicz $P_a(L,B)$
3. Oblicz $P_s(L,B)$
4. Sprawdzenie dokładności obliczeń
5. Jeżeli $\text{dokładność} > \text{dokładności_obliczeń}$ idź do pn. 4
6. Wyświetl $P_b(B,L)$
9. KONIEC

W postaci dźwiękowej:



W postaci logicznych tablic decyzji:

Pole wariantów decyzji	Kombinacje wariantów decyzji				
	2	3	4	5	
1. Ocena	1	1	1	2	
2. Zaliczone	N	T	T	T	
3. Nieobecności	1	2	3	4	
Warianty decyzji					
1. Dopuszczenie do egzaminu			X	X	
2. Warunkowe		X			
3. Nie dopuszczenie do egzaminu	X				

Prezentacja algorytmu (sposób zapisu):

Pseudokod

Przenoszenie współrzędnych (zadanie wprost):

Program przenoszenie_wspolrzednych

Zmienne

fia, laa, s, Azymut, dokladnosc, dok_obliczen : rzeczywiste

Początek

Czytaj (fia)

Czytaj (laa, s, Azymut, dokladnosc, dok_obliczen)

Fis =

Lis =

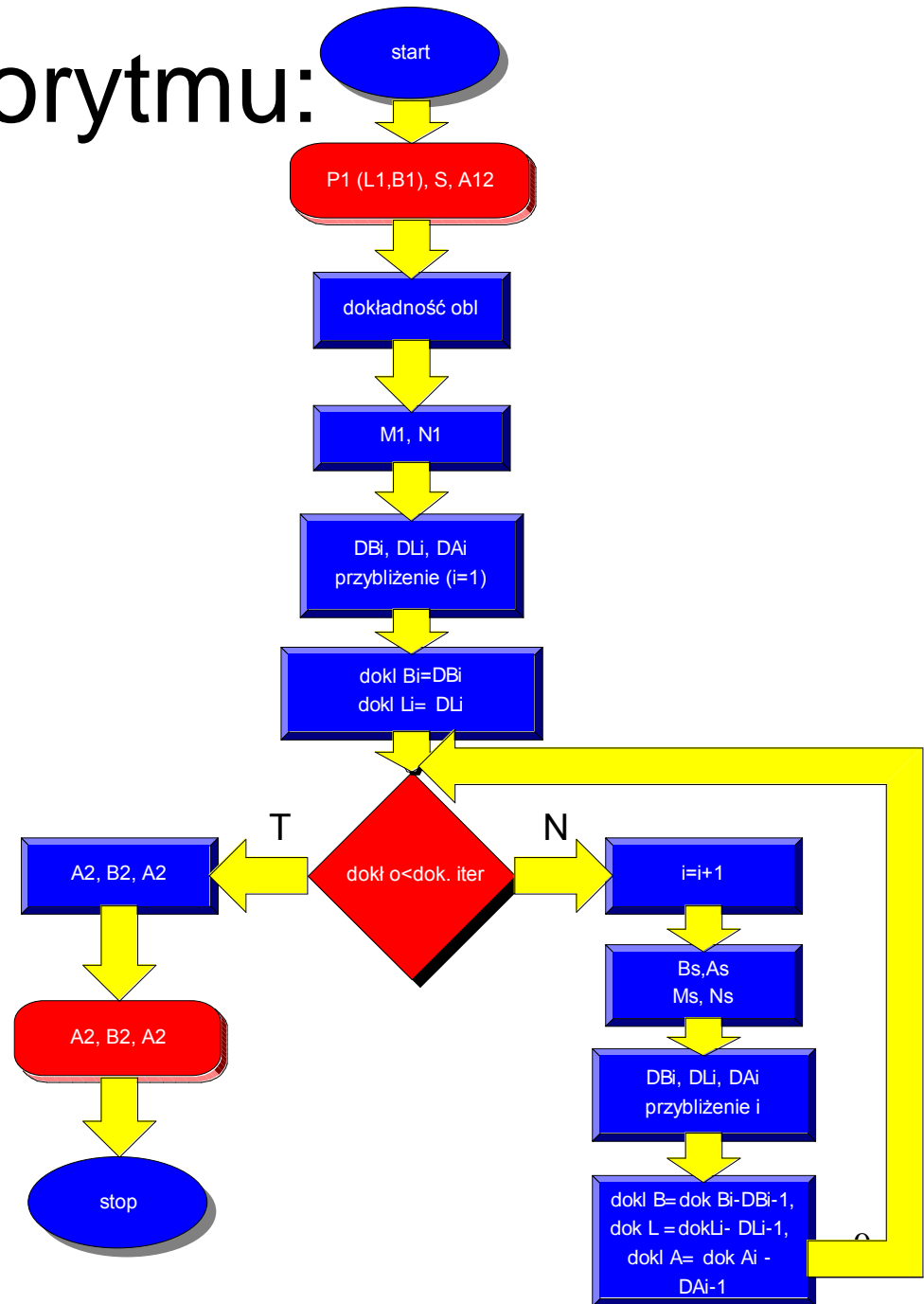
...

Pisz(wynik; fib, lab)

koniec

Prezentacja algorytmu:

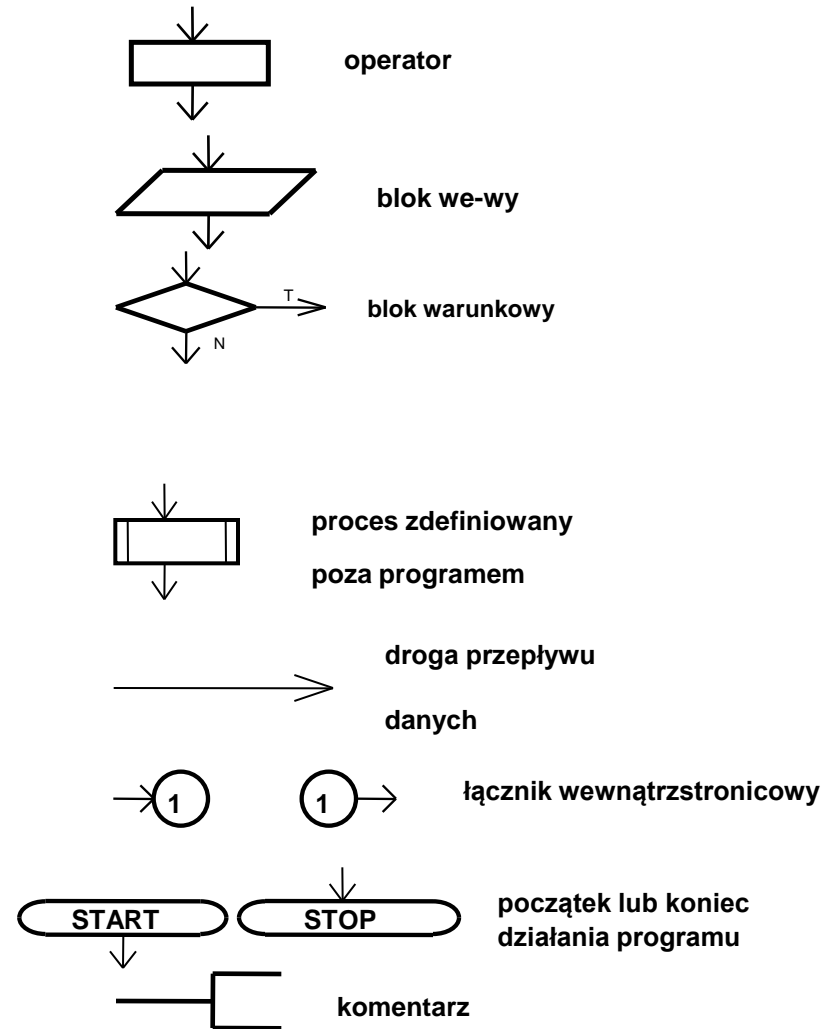
Graficzna (schemat blokowy)



1.2. Schemat blokowy

Schemat blokowy - graficzne odwzorowanie czynności programu

Elementy
schematu
blokowego



Elementy schematu blokowego

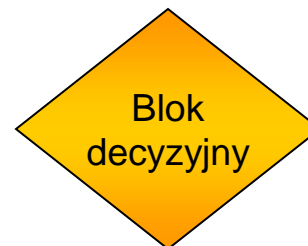
Reguły

Każdy algorytm ma jeden początek

Wszystkie drogi muszą kończyć się blokiem stop

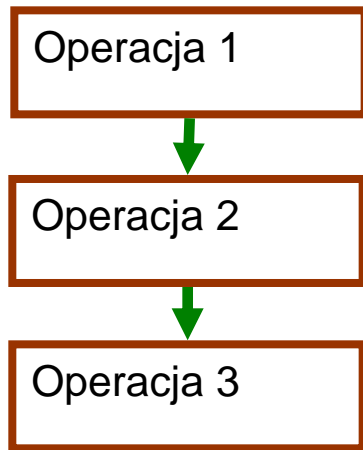
Wszystkie bloki muszą mieć zgodną z budową liczbę wejść i wyjść

Każda linia musi być opatrzona strzałką informującą o kierunku przepływu informacji

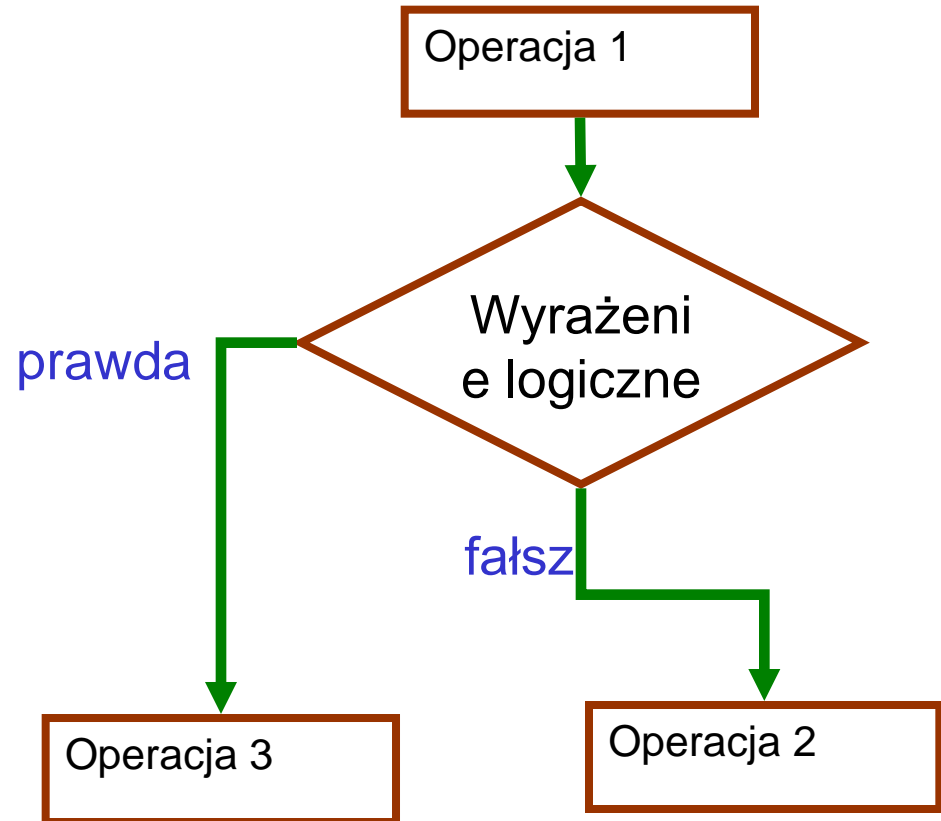


Rodzaje algorytmów przepływu informacji

Proste

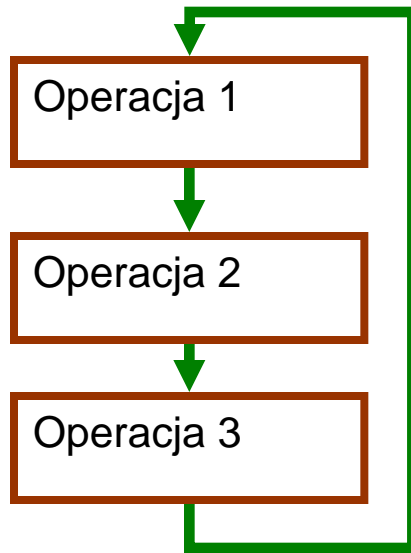


Rozgałęzione

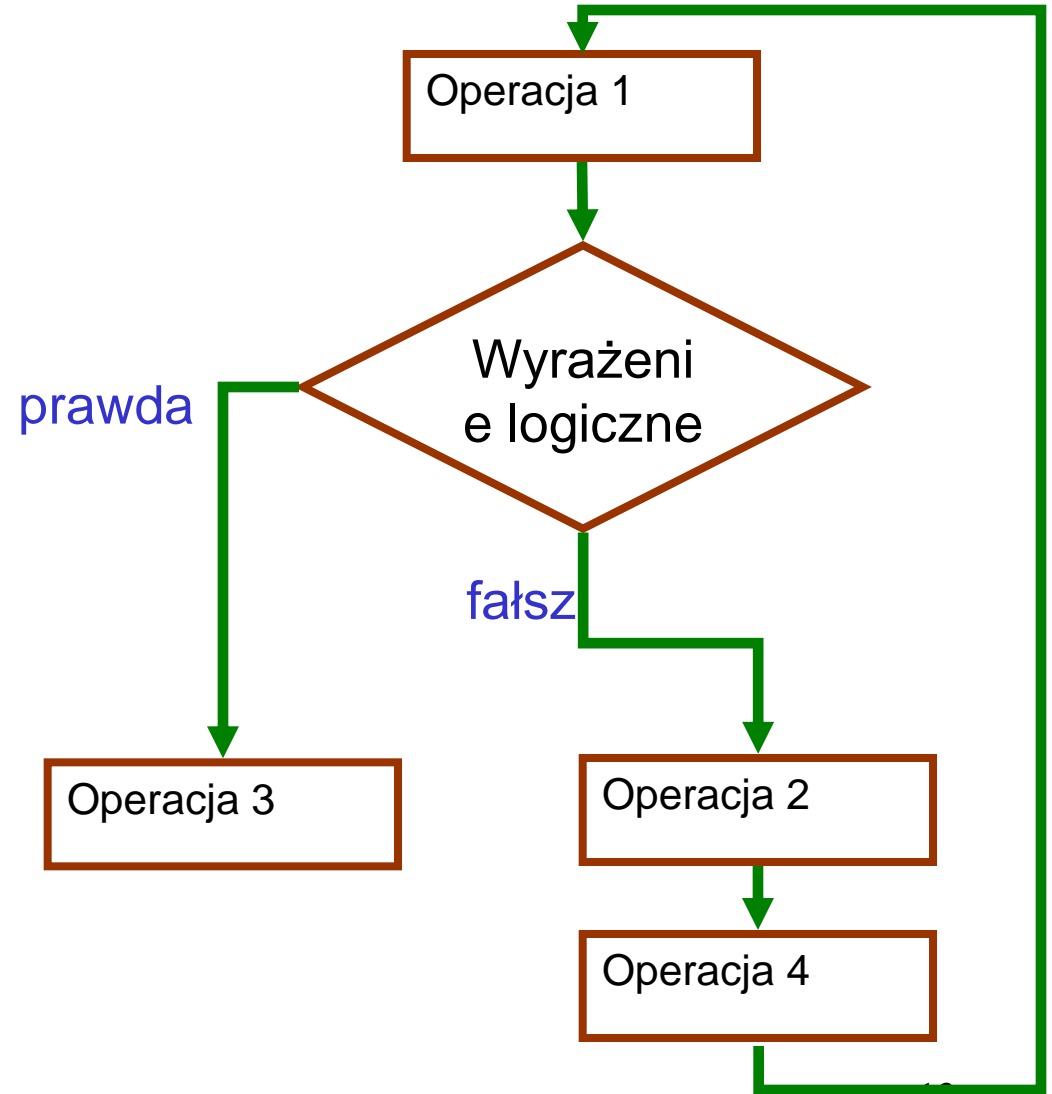


Rodzaje algorytmów przepływu informacji c.d.

Cykliczne



Mieszane



Rodzaje algorytmów

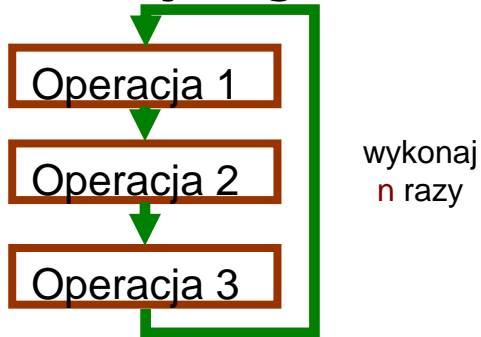
Iteracyjne

Umożliwiają wykonać wielokrotnie te same operacje

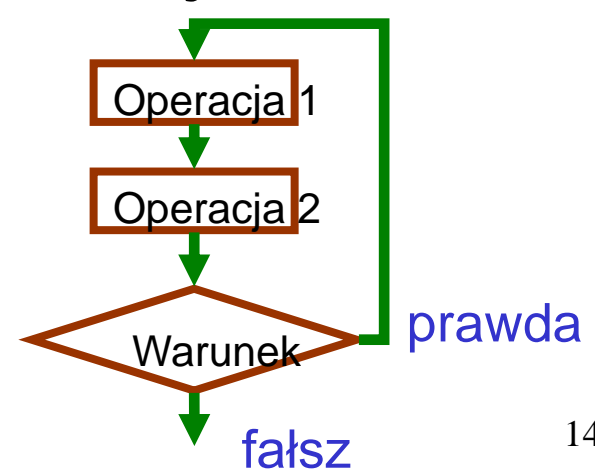
Muszą posiadać warunkowe zakończenie pętli

Często stosowana jest dekrementacja lub inkrementacja

Iteracja ograniczona



Iteracja warunkowa



Algorytmy rekurencyjne:

Właściwość podstawowa algorytmu rekurencyjnego: algorytm wywołuje sam siebie. Problemy rozwiązywane przez algorytmy rekurencyjne dają się podzielić na podproblemy, które są podobne do całości problemu.

Proces tworzenia algorytmu:

1. Zdefiniowanie pożądanego stanu wyjściowego
2. Określenie stanu wejściowego
3. Ustalenie dziedziny dopuszczalnych operacji
4. Rozbicie zadania na zadania cząstkowe
5. Budowa procedur rozwiązujących zadania cząstkowe
6. Powiązanie procedur w jedną całość
7. Prezentacja algorytmu
8. Testowanie i weryfikacja algorytmu

Schemat blokowy z rozwidleniem

Przedstaw algorytm dla poniższego zadania

Dana jest funkcja postaci

$$y(x) = \begin{cases} -x + 7 & \text{dla } x < -2 \\ x + 3 & \text{dla } -2 \leq x < 2 \\ x + 4 & \text{dla } x \geq 2 \end{cases}$$

Obliczyć wartość funkcji $y(x)$ dla zadanej wartości x

Schemat blokowy z powtórzeniami

Schemat blokowy z iteracją

Przedstaw algorytm dla poniższego zadania

Dane są macierze:

A o wymiarach $m \times n$: $a(i,j)$; $i=1, 2, \dots, m$; $j=1, 2, \dots, n$

B o wymiarach $n \times p$: $b(j,l)$; $j=1, 2, \dots, n$; $l=1, 2, \dots, p$

Obliczyć iloczyn macierzy **A** i **B**: *iloczyn_mac*.

