

Algorytm

Algorytm opisuje (krok po kroku) sposób przekształcania danych wejściowych w dane wyjściowe zgodnie z celem lub postawionym zadaniem.

Algorytm

Algorytmy pojawiły się dużo wcześniej niż pierwsze maszyny liczące albo komputery z możliwościami realizacji algorytmów w postaci programów komputerowych.

Pierwsze algorytmy tworzyli głównie matematycy. To im przede wszystkim były potrzebne zaplanowane działania w celu wykonywania skomplikowanych, jak na tamte czasy, obliczeń rachunkowych.

Nie znane było jeszcze nawet słowo *algorytm*.

Algorytm

Słowo to wywodzi się od arabskiego przydomka *al-chorezmi* („urodzony w Chorezmie”), noszonego przez matematyka, który nazywał się Muhammad ibn Musa Alchwarizmi al-Chorezmi i który żył i pracował w IX w. w Bagdadzie.

Słowo *algorytm* jest zniekształconym brzmieniem jego nazwiska.

Muhammad ibn Musa Alchwarizmi al-Chorezmi (ok. 790 – ok. 840 r. n.e.) był wybitnym matematykiem, astronomem i geografem działającym w kręgu kultury arabskiej. Uczony ten zapoczątkował i rozwinął wiele dziedzin matematyki. Wyjaśnił znaczenie używania zera i stosowania systemu pozycyjnego zapisu liczb. Dzięki jego traktatom dziesiętny system zapisu liczb rozprzestrzenił się w kręgu kultury arabskiej, a później (w XII w.) przeniknął do kultury europejskiej.

Algorytm

Za jeden z najstarszych uznaje się algorytm, który powstał przeszło 2300 lat temu.

Wymyślił go Euklides, szukając największej wspólnej miary dla dwóch odcinków. Algorytm ten, zwany powszechnie algorytmem Euklidesa, znany jest jako algorytm poszukiwania największego wspólnego dzielnika dla dwóch niezerowych liczb naturalnych (NWD).

Cechy algorytmu

- **ogólność**-co oznacza, że algorytm przeznaczony jest do rozwiązywania określonej klasy zadań, a nie tylko pojedynczego szczególnego przypadku (czyli np. rozwiązywanie równań różniczkowych 2 rzędu a nie konkretnego równania).
- **Skończoność** -możliwość uzyskania rozwiązania problemu w skończonej ilości kroków.
- **Określoność** – jednoznaczność wszystkich wykonywanych w nim operacji.
- **efektywność** - przez co rozumiemy czas potrzebny na wykonanie tego algorytmu (najczęściej liczony w umownych jednostkach np. ilości wykonywanych operacji w zależności od danych wejściowych) lub zapotrzebowanie algorytmu na pamięć.

Etapy rozwiązywania problemów:

1. Sformułowanie zadania
2. Określenie danych wejściowych
3. Określenie celu, czyli wyniku
4. Poszukiwanie metody rozwiązania, czyli algorytmu
5. Przedstawienie algorytmu w postaci:
 - opisu słownego
 - listy kroków
 - schematu blokowego
 - jednego z języków programowania
6. Analiza poprawności rozwiązania
7. Testowanie rozwiązania dla różnych danych
ocena efektywności przyjętej metody

Algorytm

Pojęcie „algorytm” swoją popularność i rozpowszechnienie zawdzięcza przede wszystkim informatyce. A powodem są komputery. W bardzo uproszczony sposób, związek algorytmów z komputerami, czyli pośrednio z informatyką, można ująć następująco – komputery głównie wykonują programy, a każdy program jest zapisem jakiegoś algorytmu.

Algorytmy można zapisywać na papierze, na tablicy lub opisać słownie. Tak zresztą często postępujemy również na lekcjach informatyki. Ale na tym nie koniec. Dopiero uzmysłowienie sobie, że ten przepis ma być wykonany przez komputer czyni z niego twór informatyczny. Najlepiej ujmuje ten fakt powiedzenie, którego autorem jest jeden z największych żyjących informatyków, Donald E. Knuth:

Mówi się często, że człowiek dotąd nie zrozumie czegoś,
zanim nie nauczy tego – kogoś innego.

W rzeczywistości,
człowiek nie zrozumie czegoś naprawdę,
zanim nie zdoła nauczyć tego – komputera.

Donald Ervin Knuth

Donald Ervin Knuth (ur. 10 stycznia 1938 r. w Milwaukee) – amerykański matematyk, informatyk, emerytowany profesor na katedrze informatyki Uniwersytetu Stanforda. Jeden z pionierów informatyki – jest najbardziej znany z wielotomowego dzieła Sztuka programowania (The Art of Computer Programming), uznawanego za najbardziej dogłębne, chociaż na razie niedokończone opracowanie na temat analizy algorytmów. Jest też autorem systemu składu drukarskiego TeX i języka opisu fontów METAFONT . (Wikipedia)

Algorytm i specyfikacja

Algorytm, to tak skonstruowany przepis, że może go zrozumieć i wykonać komputer.

Oczywiście do wszystkiego jest potrzebny człowiek, który ma przygotować i podać ten przepis komputerowi.

Stąd, ze względu na wykonawcę-komputer, o algorytmie mówi się m.in., że składa się z uporządkowanych kroków, jednoznacznie opisanych, wykonalnych za pomocą prostych i znanych operacji, ma skończone działanie itd. Dodatkowo, opis kroków algorytmu powinien być poprzedzony specyfikacją, czyli precyzyjnym opisem wejścia i wyjścia, a więc danych i wyników. To wszystko po to, by siadając do tłumaczenia komputerowi, co ma zrozumieć i wykonać, ani tłumaczący, ani później komputer nie mieli żadnych wątpliwości, o co chodzi.

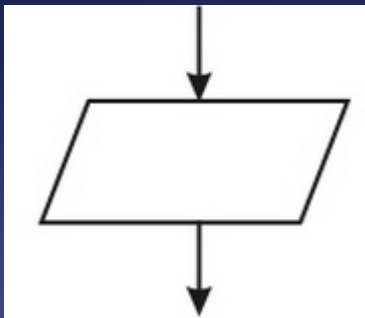
Maciej M. Sysło
Instytut Informatyki, Uniwersytet
Wrocławski

Schemat blokowy

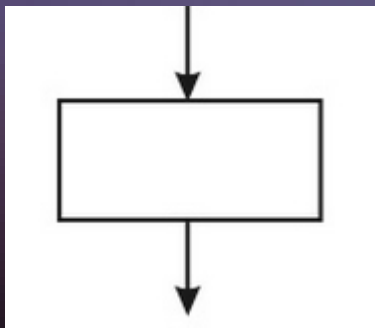
Schemat blokowy (flowchart) stosowany jest do przedstawienia algorytmów w postaci diagramu reprezentującego kolejne wykonywane czynności w algorytmie. Odpowiednie figury geometryczne reprezentują różne rodzaje operacji na danych i są łączone liniami zgodnie z kolejnością ich wykonywania wg realizowanego algorytmu



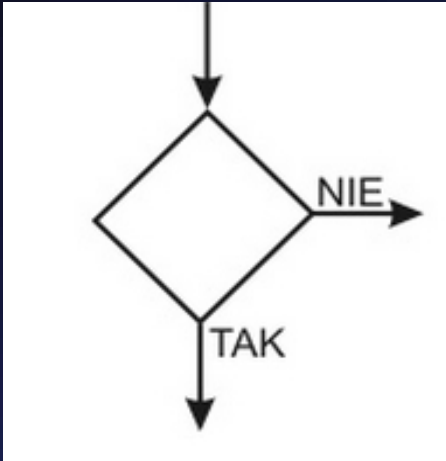
blok graniczny - blok startu, końca programu;



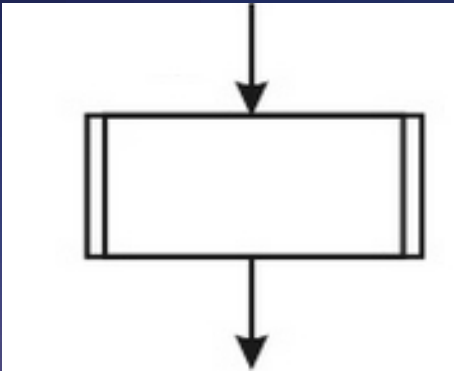
blok wejścia-wyjścia - przedstawia czynność wprowadzania danych do programu i przyporządkowania ich wartości zmiennym dla późniejszego ich wykorzystania, jak również wyprowadzenia wyników obliczeń



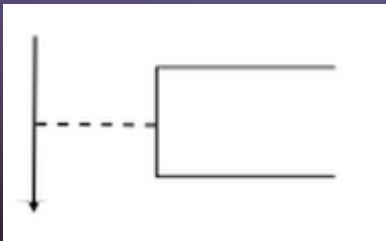
blok operacyjny



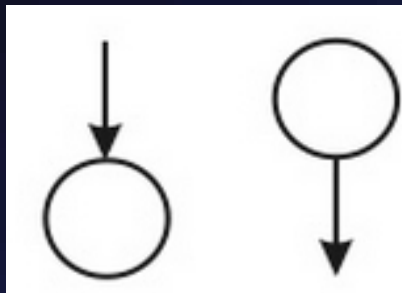
blok decyzyjny/warunkowy



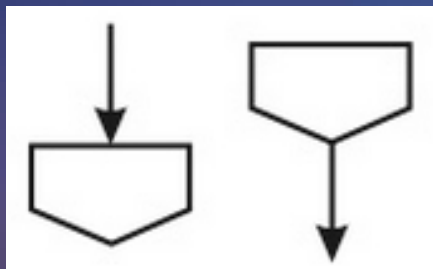
**blok fragmentu programu -
przedstawia część programu
zdefiniowanego odrębnie, np.
sortowanie**



blok komentarza



łącznik wewnętrzny - służy do łączenia odrębnych części schematu znajdujących się na tej samej stronie, powiązane ze sobą łączniki oznaczone są tym samym napisem, np. A1, A2, B7 itp.



łącznik zewnętrzny - służy do łączenia odrębnych części schematu znajdujących się na odrębnych stronach, powinien być opisany jak łącznik wewnętrzny, poza tym powinien zawierać numer strony, do której się odwołuje, np. 4.A3, 2.B2 itp.

a) **Algorytm liniowy (sekwencyjny)** – kolejność wykonywanych czynności nie jest zależna od wartości danych wejściowych

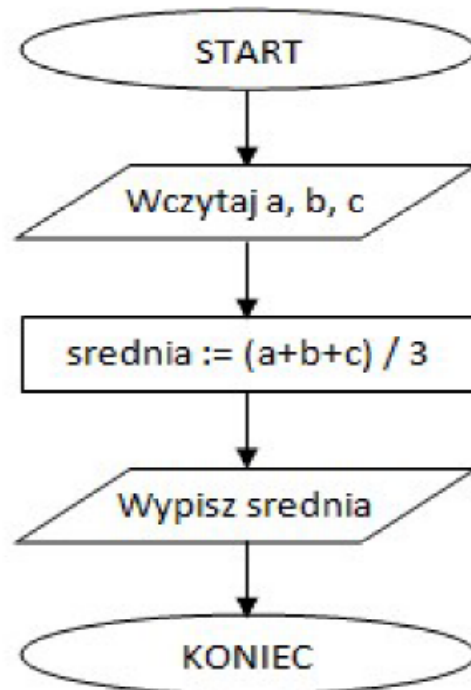
Przykład :

Algorytm obliczający średnią arytmetyczną podanych liczb:

Dane wejściowe: trzy liczby **a**, **b**, **c**.

Dane wyjściowe: liczba **srednia**, będąca średnią arytmetyczną tych liczb.

1. Start
2. Wczytaj **a**, **b**, **c**
3. **srednia:=(a+b+c)/3**
4. Wypisz **srednia**
5. Koniec.



Rodzaje algorytmów

b) **Algorytm z rozgałęzieniami (warunkowy)**– kolejność wykonywania operacji zależy od wartości danych wejściowych

Przykład :

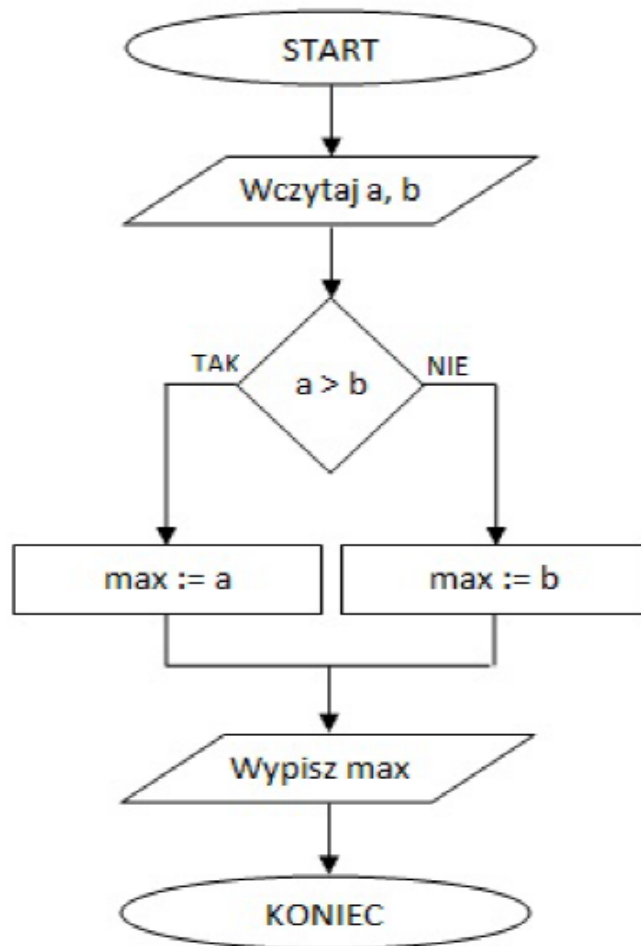
Algorytm zwracający wartość maksymalną z dwóch wprowadzonych wartości

Rodzaje algorytmów

Dane wejściowe: dwie liczby **a**, **b**.

Dane wyjściowe: liczba **max**, będąca wartością maksymalną.

1. Start
2. Wczytaj **a**, **b**
3. Jeśli **a > b** to **max := a**, w przeciwnym razie **max := b**
4. Wypisz **max**
5. Koniec.

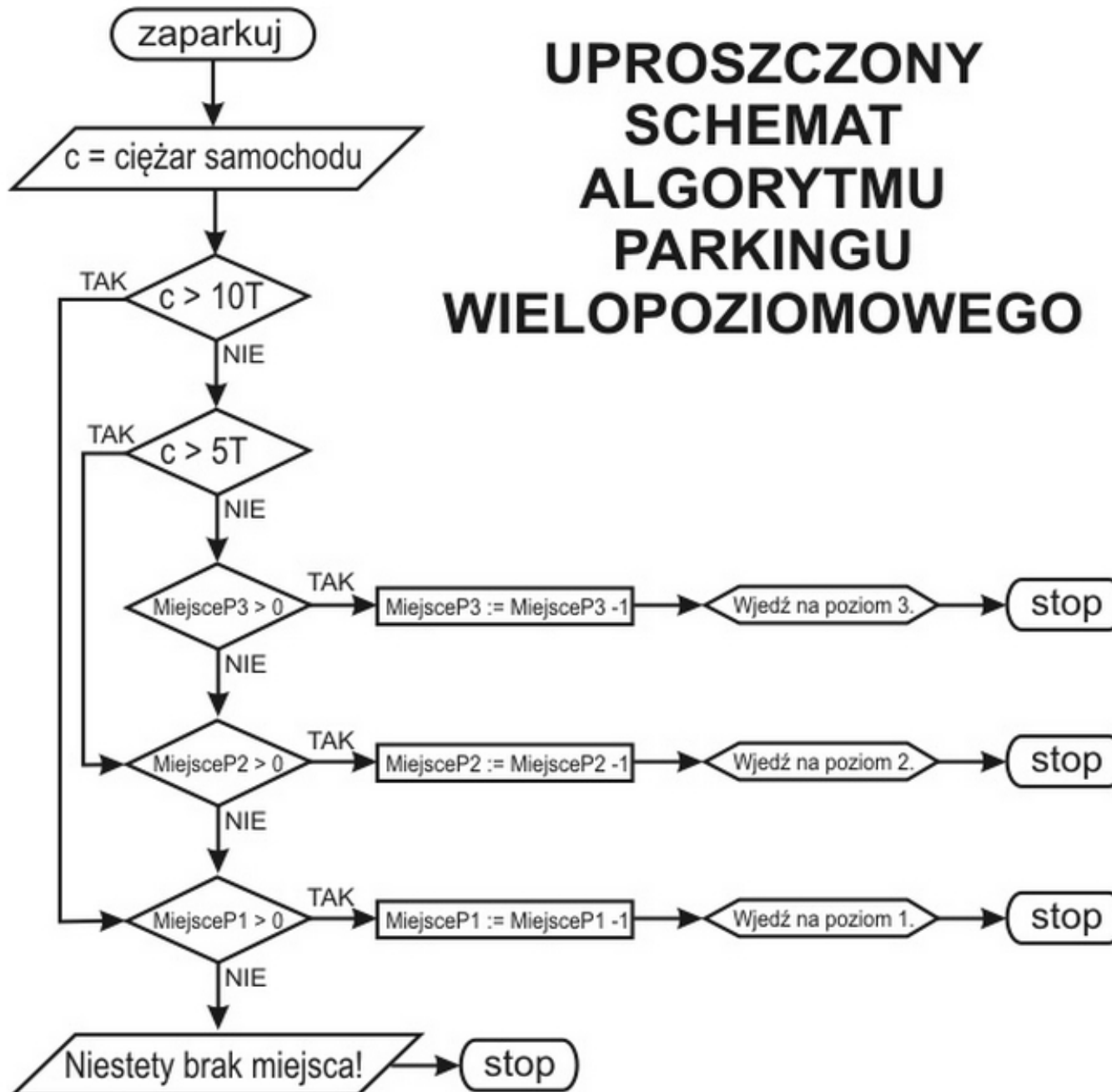


Rodzaje algorytmów

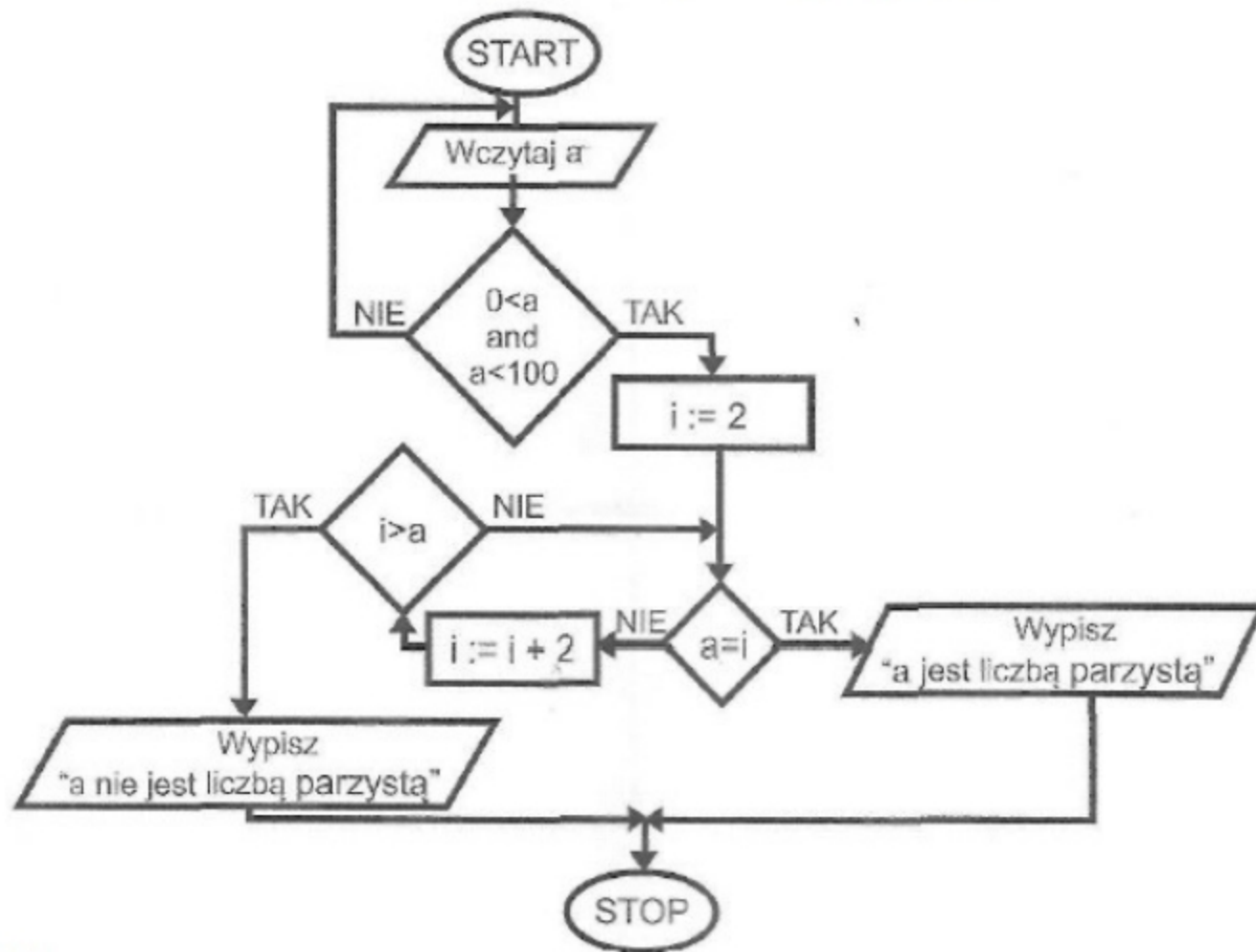
c) **Algorytm iteracyjny**-zawiera instrukcje, które nakazują wielokrotne powtarzanie pewnych czynności. Iteracje występują w dwóch podstawowych odmianach: iteracja z określoną liczbą powtórzeń : wykonuj czynność dokładnie N razy, iteracja warunkowa: wykonaj czynność, dopóki jest spełniony warunek. Algorytm iteracyjny może działać na danych o dowolnej długości (wielkości).

d) **Algorytm rekurencyjny**- Charakterystyczną cechą funkcji (procedury) rekurencyjnej jest to, że wywołuje ona samą siebie.

UPROSZCZONY SCHEMAT ALGORYTMU PARKINGU WIELOPOZIOMOWEGO



Rysunek przedstawia schemat blokowy, na którym znajdują się dwa bloki



- A. decyzyjne.
- B. operacyjne.
- C. warunkowe.
- D. wprowadzania danych.

Cechy algorytmu

- 1) poprawność (algorytm daje dobre wyniki, odzwierciedlające rzeczywistość)
- 2) jednoznaczność (brak rozbieżności wyników przy takich samych danych, jednoznaczne opisanie każdego kroku)
- 3) skończoność (wykonuje się w skończonej ilości kroków)
- 4) sprawność (czasowa - szybkość działania i pamięciowa – "zasobożerność")
- 5) prostota wykonania (operacje powinny być jak najprostsze)