

# WSTĘP DO INFORMATYKI

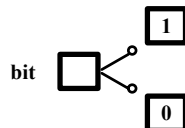
**Program wykładu na studiach dziennych:**

- |                                |                                 |
|--------------------------------|---------------------------------|
| 1. Wprowadzenie do algorytmiki | 8. Złożoność problemów alg.     |
| 2. Struktura algorytmu         | 9. Problemy rozstrzygalne i nie |
| 3. Struktury danych            | 10. Współbieżność               |
| 4. Język programowania         | 11. Algorytmy probabilistyczne  |
| 5. Metody algorytmiczne        | 12. Algorytmy heurystyczne      |
| 6. Poprawność algorytmów       | 13. Wykorzystanie złożoności    |
| 7. Złożoność algorytmów        | 14. Kierunek „inteligencja”     |

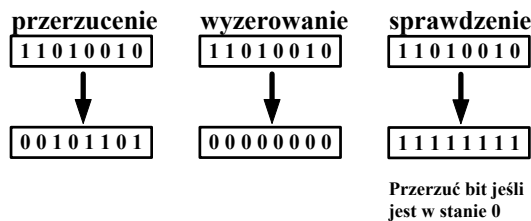
**Literatura:**

- D.Harel „Rzecz o istocie informatyki. Algorytmika” WNT (2000)
- A.J.Aho, J.E.Hopcroft, J.D.Ullman „Struktury danych i algorytmy” PWN (1983)
- A.J.Aho, J.E.Hopcroft, J.D.Ullman „Projektowanie i analiza algorytmów komputerowych” PWN (1988)
- N.Wirth „Algorytmy + struktury danych = programy” WNT (1989) wyd. 2
- T.Cormen, C.Leiserson, R.Rivest „Wprowadzenie do algorytmów” WNT (1997)
- E.W. Dijkstra „Umiejętność programowania” WNT 1985 (wyd. 2)
- P.Wróblewski „Algorytmy. Struktury danych i techniki programowania” Helion 1997 (wyd. 2)

KOMPUTER = ZESTAW PRZEŁĄCZNIKÓW



Podstawowe operacje na bitach:



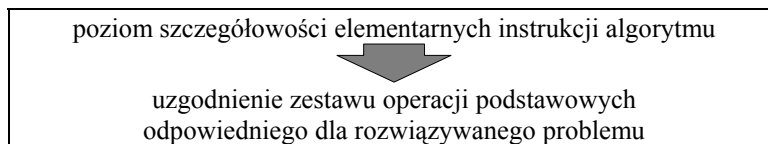
Funkcjonalny model sterowania procesem (komputerem):



**Prekursorzy:**

- Euklides (400 - 300 rok p.n.e) - matematyk, autor algorytmu wyznaczania NWD dwóch liczb
- Muhammed Alchwarizmi (IX w.) - matematyk, autor reguł podstawowych operacji arytmetycznych dla liczb dziesiętnych (łac. Algorismus)
- Joseph Jacquard (1801) - wynalazca krosna tkackiego sterowanego dziurkowanymi kartami

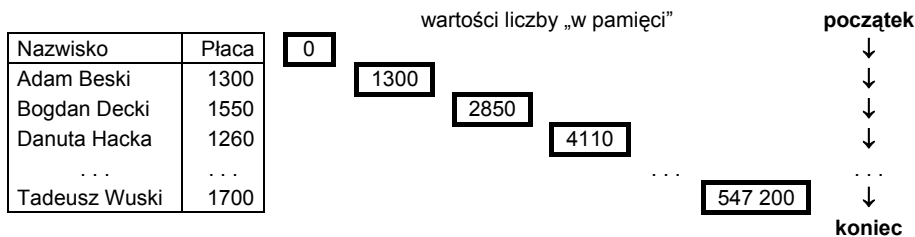
- Charles Babbage (1833) - matematyk, wynalazca „maszyny różnicowej” i autor projektu „maszyny analitycznej” sterowanej algorytmami kodowanymi na dziurkowanych kartach
- Augusta Ada King - matematyczka, „programistka” maszyny Babbage
- Herman Hollerith (1890) - wynalazca maszyny wspomagającej spis powszechny w USA.
- Pierwsze komputery elektroniczne - lata 40 XX w. (Alan Turing, Howard Aiken, John Mauchly, John von Neumann)
- Informatyka staje się dyscypliną akademicką w połowie lat 60 - Association for Computing Machinery (ACM) publikuje zalecenia programowe dla prowadzenia wykładów



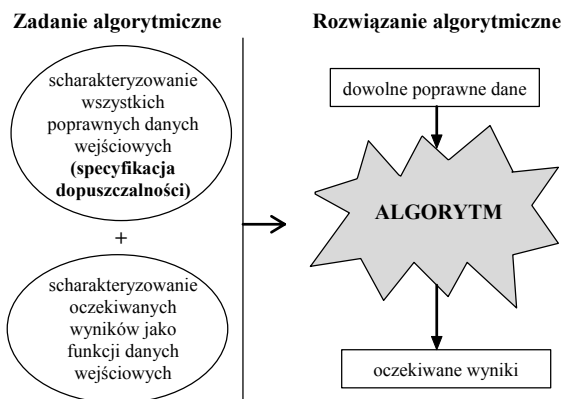
Krótkie algorytmy mają opisywać także długotrwałe procesy!

*Sumowanie zarobków pracowników przedsiębiorstwa*

1. zanotuj „w pamięci” liczbę 0;
2. przeglądaj kolejno kwestionariusze pracowników, dodając zarobki każdego z nich do liczby „w pamięci”
3. kiedy osiągniesz koniec listy, podaj wartość liczby „w pamięci” jako wynik



Co to jest zadanie algorytmiczne i jego rozwiązanie:



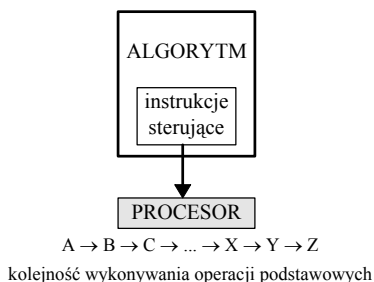
Podstawowe założenie, którego spełnienie jest warunkiem koniecznym dla uznania operacji za podstawową:

**czas wykonania każdej z operacji podstawowych jest skończony**

Wyzwanie dla algorytmiki to złożoność rzeczywistych zadań algorytmicznych, np.:

1. *dane wejściowe - poprawna sytuacja szachowa*  
*oczekiwany wynik - najlepszy ruch białych*
2. *2000000 egz. gazet należy rozwieźć do 10000 punktów sprzedaży w 100 miastach mając do dyspozycji 50 samochodów i 60 kierowców, tak aby przejechać minimalną liczbę kilometrów.*

Rola algorytmu w sterowaniu wykonaniem zadania przez procesor:



Struktury (instrukcje) sterujące:

- **bezpośrednie następstwo** - „wykonaj instrukcję A, a potem B”
- **wybór warunkowy** - „jeśli warunek logiczny Q jest spełniony, to wykonaj instrukcję A, w przeciwnym razie wykonaj B”
- **iteracje** (pętle) w różnych odmianach np.
  - iteracja warunkowa** - „dopóki warunek logiczny Q jest spełniony, wykonaj instrukcję A”
  - iteracja warunkowa** - „wykonuj instrukcję A, aż do spełnienia warunku logicznego Q”
  - iteracja ograniczona** - „wykonaj instrukcję A dokładnie N razy”

A i B to operacje podstawowe lub całe fragmenty (bloki) algorytmu. Q to zdanie logiczne, którego wartość (prawda lub fałsz) można określić w momencie przejścia do wykonania warunkowej instrukcji sterującej.

Sumowanie zarobków N pracowników przedsiębiorstwa

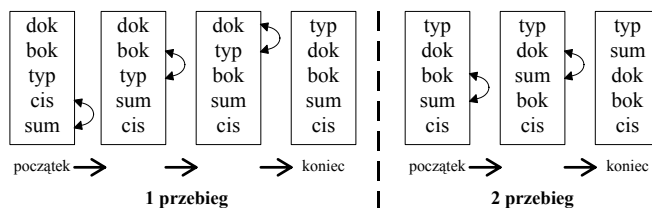
1. zainicjuj „w pamięci” liczbę 0; wskaż na pierwszą płacę z listy;
2. wykonaj co następuje N - 1 razy:
  - 2.1. dodaj wskazaną płacę do liczby „w pamięci”;
  - 2.2. wskaż na następną płacę;
3. dodaj wskazaną płacę do liczby „w pamięci”;
4. podaj wartość liczby „w pamięci” jako wynik.

**Struktury sterujące mogą być zagnieżdżane jedna w drugiej**

- np. iteracje
1. wykonaj co następuje dokładnie N razy (*iteracja zewnętrzna*)
    - 1.1. dopóki Q wykonuj A (*iteracja wewnętrzna*)

Sortowanie bąbelkowe:

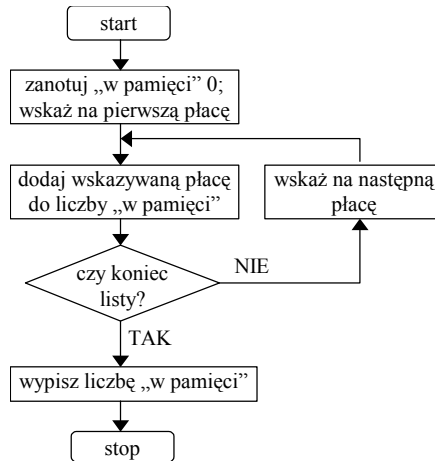
dane wejściowe - porządek i nieuporządkowana lista N elementów  
 wynik - uporządkowana lista N elementów



1. wykonaj co następuje N - 1 razy:
  - 1.1. wskaż na pierwszy element listy;
  - 1.2. wykonaj co następuje N - 1 razy:
    - 1.2.1. porównaj wskazany element z elementem następnym;
    - 1.2.2. jeśli porównywane elementy są w niewłaściwej kolejności, zamień je miejscami;
    - 1.2.3. wskaż następny element.

Kontrowersyjna jest instrukcja **skoku** - „skocz do wskazanego miejsca w algorytmie”

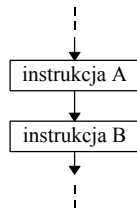
**Schematy blokowe algorytmów**  
*Sumowanie płac pracowników:*



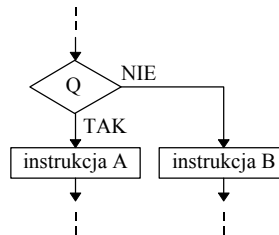
Schematy blokowe podstawowych instrukcji sterujących:

- **bezpośrednie następstwo** - „wykonaj *A*, a potem *B*”.
- **wybór warunkowy** - „jeśli *Q*, to wykonaj *A*, w przeciwnym razie wykonaj *B*”
- **iteracja ograniczona** - „wykonaj *A* dokładnie *N* razy”,
- **iteracja warunkowa** - „dopóki *Q*, wykonuj *A*” lub „wykonuj *A* aż do *Q*”

**bezpośrednie następstwo**

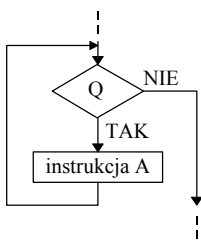


**wybór warunkowy**

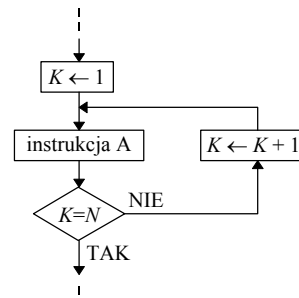
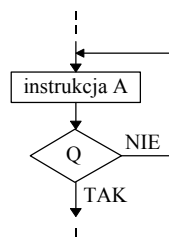


**iteracja ograniczona**

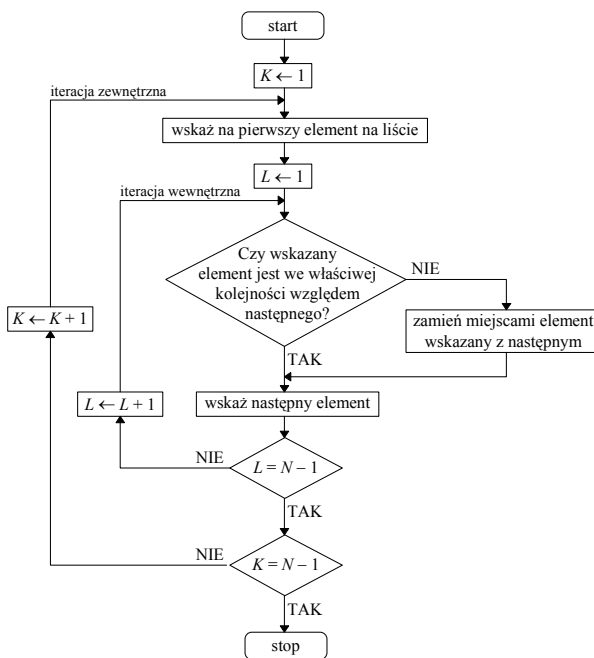
**iteracja warunkowa „dopóki”**



**iteracja warunkowa „aż do”**



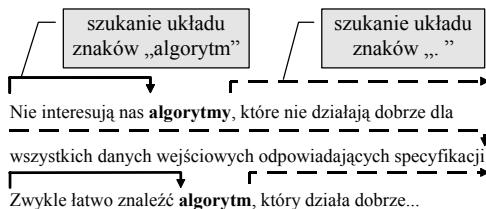
Schemat blokowy sortowania bąbelkowego:



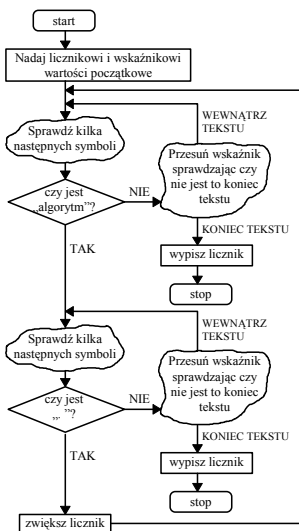
**Podprogramy, czyli procedury**

Przykład zastosowania w problemie przeszukiwania tekstu:

*dane wejściowe* - tekst w języku polskim  
*wynik* - liczba zdań zawierających słowo „algorytm”

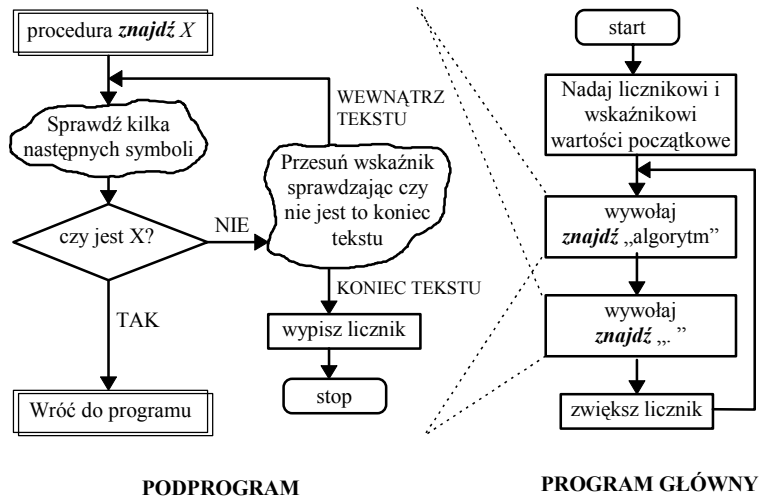


Zarys schematu blokowego:



procedura *znajdź X*:

1. wykonuj co następuje, aż będzie wskazana układ znaków *X* albo osiągnięty zostanie koniec tekstu:
  - 1.1. przesuń wskaźnik w tekście o jeden znak do przodu;
2. jeśli został osiągnięty koniec tekstu, wypisz wartość licznika i zatrzymaj się;
3. w przeciwnym razie wróć do głównego programu



Stosowanie procedur pozwala na:

- oszczędne zapisywanie tekstu źródłowego programu (unikamy wielokrotnego powtarzania strukturalnie takich samych fragmentów),
- uzyskanie zapisu czytelnie przedstawiającego strukturę programu (część struktur sterujących wchodzących w zakres procedury nie jest widoczna na poziomie programu głównego),
- „analityczne” budowanie algorytmu (najpierw opracowujemy jego fragmenty w postaci procedur, z których następnie składamy program główny),
- „syntetyczne” budowanie algorytmu (składamy algorytm z procedur wcześniej opracowanych),
- budowanie algorytmów rekurencyjnych (jeśli można wywoływać procedurę w niej samej).