

# Algorytmy

AN-004  
AN-005

# CZYM JEST ALGORYTM?

To skończony ciąg jasno zdefiniowanych kroków potrzebnych do wykonania danego zadania. Typowym przykładem z życia codziennego może być na przykład przepis kuchenny.

# Cechy algorytmu

- **Poprawność** (algorytm daje oczekiwane wyniki)
- **Jednoznaczność** (zawsze daje te same wyniki, przy takich samych danych wejściowych)
- **Skończoność** (wykonuje się w skończonej liczbie kroków)
- **Sprawność** (czasowa - szybkość działania i pamięciowa)

# RODZAJE ALGORYTMÓW

- Liniowe ( sekwencyjne )
- Warunkowe
- Iteracyjne
- Rekurencyjne

# Algorytm liniowy (sekwencyjny)

Algorytm liniowy (sekwencyjny) – instrukcje wykonywane są po kolei (w porządku, w jakim zostały wprowadzone).

Przykład algorytmu liniowego – przepis na jajecznicę.

1. Przygotuj jajko, tłuszcz, przyprawy
2. Wrzuć tłuszcz na patelnię
3. Podgrzej patelnię
4. Wbij jajko
5. Dodaj przyprawy
6. Wymieszaj
7. Wyłóż na talerz

# Algorytmy z rozgałęzieniami

Algorytmy z rozgałęzieniami to coś podobnego do algorytmów liniowych, z tym, że są różne drogi (rozgałęzienia) do uzyskania wyniku - inaczej mówiąc, algorytm wykonywany jest w zależności od pewnych warunków.

## Przykład algorytmu z rozgałęzieniami –

„Sprawdź czy pada deszcz, jeśli tak, weź kurtkę z kapturem”

1. Wyrzyj przez okno
2. Czy pada deszcz?
  - a) jeśli tak, przejdź do kroku 3
  - b) Jeśli nie, przejdź do kroku 4
- 3 Weź kurtkę z kapturem
- 4 Wyjdź

# Algorytm iteracyjny

Rodzaj algorytmu w którym wielokrotnie wykonuje się pewne instrukcje, dopóki nie zostanie spełniony określony warunek.

**Przykład algorytmu iteracyjnego –**

Sprawdzaj co „jakiś czas” jakie jest światło,  
jeśli jest czerwone czekaj, jeśli jest zielone przejdź

1. Stój
2. Czy świeci się czerwone światło?
  - A. Jeśli TAK to przejdź do kroku 1
  - B. Jeśli NIE to przejdź do kroku 3
3. Można przejść przez ulicę zachowując ostrożność

# Algorytm rekurencyjny

Algorytm rekurencyjny to taki, który w pewnym kroku zawiera polecenie wykonania siebie samego dla innych danych.

Przykład rekurencji.

Algorytm „jedz kaszkę”

1. Jedz kaszkę (weź łyżkę kaszki i połknij)
2. Jedz kaszkę dalej, aż talerz będzie pusty



# SPOSÓB ZAPISU ALGORYTMU

- Opis słowny
- Lista kroków
- Schemat blokowy

# Opis słowny

Jeżeli naszym zadaniem jest przygotowanie kawy to możemy zadanie to przedstawić jako opis kolejnych czynności:

**Najpierw przygotuj wszystkie potrzebne składniki. Następnie wsyp kawę do filiżanki. Zagotuj wodę i zalej kawę wrzącą wodą. Dodaj cukru do smaku i mleka jeżeli lubisz. Zamieszaj kawę i gotowe**

# Lista kroków

Poprzedni opis przygotowania kawy jest dość ogólny i nie uwzględnia wszystkich zmiennych. Zapisując go w postaci listy dokładnie określonych kroków i dodając kilka warunków otrzymamy dużo doskonalszy algorytm:

# Lista kroków

1. start
2. przygotuj składniki
3. wsyp kawę do filiżanki
4. zagotuj wodę (2 minuty)
5. jeśli mam gorącą wodę przejdź do kroku 6, jeśli nie przejdź do 4
6. zalej kawę gorącą wodą
7. dodaj cukier, jeśli nie słodzisz przejdź do kroku 8
8. dodaj śmietankę, jeśli bez śmietanki przejdź do kroku 9
9. pomieszaj
10. stop

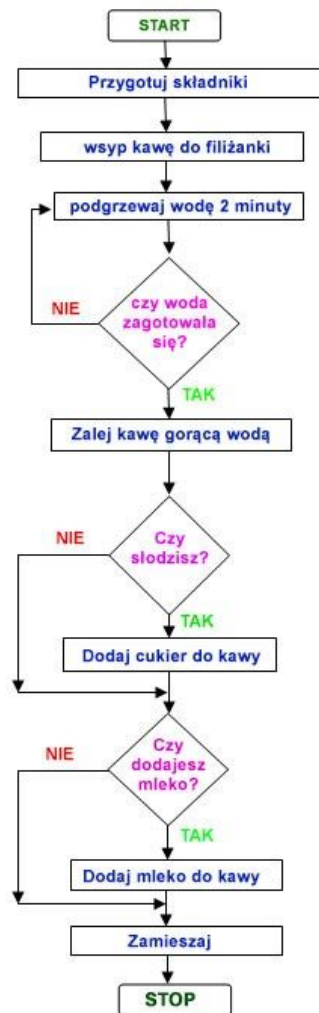
# Schemat blokowy

Do zapisu prostych algorytmów, które mają być wykonywane przez człowieka możemy używać języka potocznego. W przypadku algorytmów bardziej skomplikowanych ten zapis będzie jednak nieczytelny i nie sprawdzi się. Dlatego używa się bardziej przejrzystego sposobu zapisu algorytmów  
- schematu blokowego.

# Schemat blokowy

Schemat blokowy to graficzny zapis algorytmu rozwiązania zadania, przedstawiający opis i kolejność wykonywania czynności realizujących dany algorytm.

Algorytm parzenia kawy zapisany w postaci schematu blokowego wygląda następująco:



# Algorytmy sortujące

**Przykładem algorytmu sortującego jest algorytm sortowania bąbelkowego. Algorytm ten jest jednym z najstarszych algorytmów sortujących. Zasada jego działania opiera się na cyklicznym porównywaniu par sąsiadujących elementów i zamianie ich kolejności w przypadku niespełnienia kryterium porządkowego zbioru. Operację tę wykonujemy dotąd, aż cały zbiór zostanie posortowany.**

# Sortowanie bąbelkowe

## Przykład działania

Ciąg wejściowy [ 4, 2, 5, 1, 7 ] Każdy wiersz symbolizuje wypchnięcie kolejnego największego elementu na koniec („wypłynięcie największego bąbelka”). Niebieskim kolorem oznaczono końcówkę ciągu już posortowanego.

$[4, 2, 5, 1, 7] \rightarrow [2, 4, 5, 1, 7] \rightarrow [2, 4, 5, 1, 7] \rightarrow [2, 4, 1, 5, 7]$   
 $\underbrace{4 > 2} \quad \underbrace{4 < 5} \quad \underbrace{5 > 1} \quad \underbrace{5 < 7}$

$[2, 4, 1, 5, 7] \rightarrow [2, 4, 1, 5, 7] \rightarrow [2, 1, 4, 5, 7]$   
 $\underbrace{2 < 4} \quad \underbrace{4 > 1} \quad \underbrace{4 < 5}$

$[2, 1, 4, 5, 7] \rightarrow [1, 2, 4, 5, 7]$   
 $\underbrace{2 > 1} \quad \underbrace{2 < 4}$

$[1, 2, 4, 5, 7]$   
 $\underbrace{1 < 2}$



# Sortowanie bąbelkowe

Kod algorytmu w języku python

```
data = [4, 2, 5, 1, 7]
```

```
def sort(data):
```

```
    for i in range(len(data) - 1, 0, -1):
```

```
        for j in range(i):
```

```
            if data[j] > data[j + 1]:
```

```
                data[j], data[j + 1] = data[j + 1], data[j]
```

```
sort(data)
```

```
print(data)
```



# DZIĘKUJEMY ZA UWAGĘ

# Źródło

- › <https://sites.google.com/site/pchalgorytmika>
- › [https://pl.wikipedia.org/wiki/Sortowanie\\_babelkowe](https://pl.wikipedia.org/wiki/Sortowanie_babelkowe)
- › [https://pl.wikibooks.org/wiki/Kody\\_źródłowe/Sortowanie\\_babelkow](https://pl.wikibooks.org/wiki/Kody_źródłowe/Sortowanie_babelkow)
- › <https://www.jagiello-debica.edu.pl/index.php/120-kat-infa/zlgorytmy/299-definicja-i-cechy-algorytmu>