

# Rodzaje sortowania

sortowanie bąbelkowe (ang. bubblesort) –  $O(n^2)$

sortowanie przez wstawianie (ang. insertion sort) –  $O(n^2)$

sortowanie przez scalanie (ang. merge sort) –  $O(n \log n)$ , wymaga  $O(n)$  dodatkowej pamięci

sortowanie przez zliczanie (ang. counting sort lub count sort) –  $O(n + k)$ , wymaga  $O(n + k)$  dodatkowej pamięci

sortowanie kulekowe (ang. bucket sort) –  $O(n)$ , wymaga  $O(k)$  dodatkowej pamięci

sortowanie pozycyjne (ang. radix sort) –  $O(d(n + k))$ , gdzie  $k$  to wielkość domeny cyfr, a  $d$  szerokość kluczy w cyfrach. Wymaga  $O(n + k)$  dodatkowej pamięci

sortowanie biblioteczne (ang. library sort) –  $O(n \log n)$ , pesymistyczny  $O(n^2)$

## Złożoność obliczeniowa

Ilość zasobów niezbędnych do wykonania algorytmu można rozumieć, jako jego złożoność. W zależności od rozważanego zasobu mówimy o złożoności czasowej czy też złożoności pamięciowej. Oczywiście w większości wypadków ilość potrzebnych zasobów będzie się różnić w zależności od danych wejściowych z zakresu danego zagadnienia.

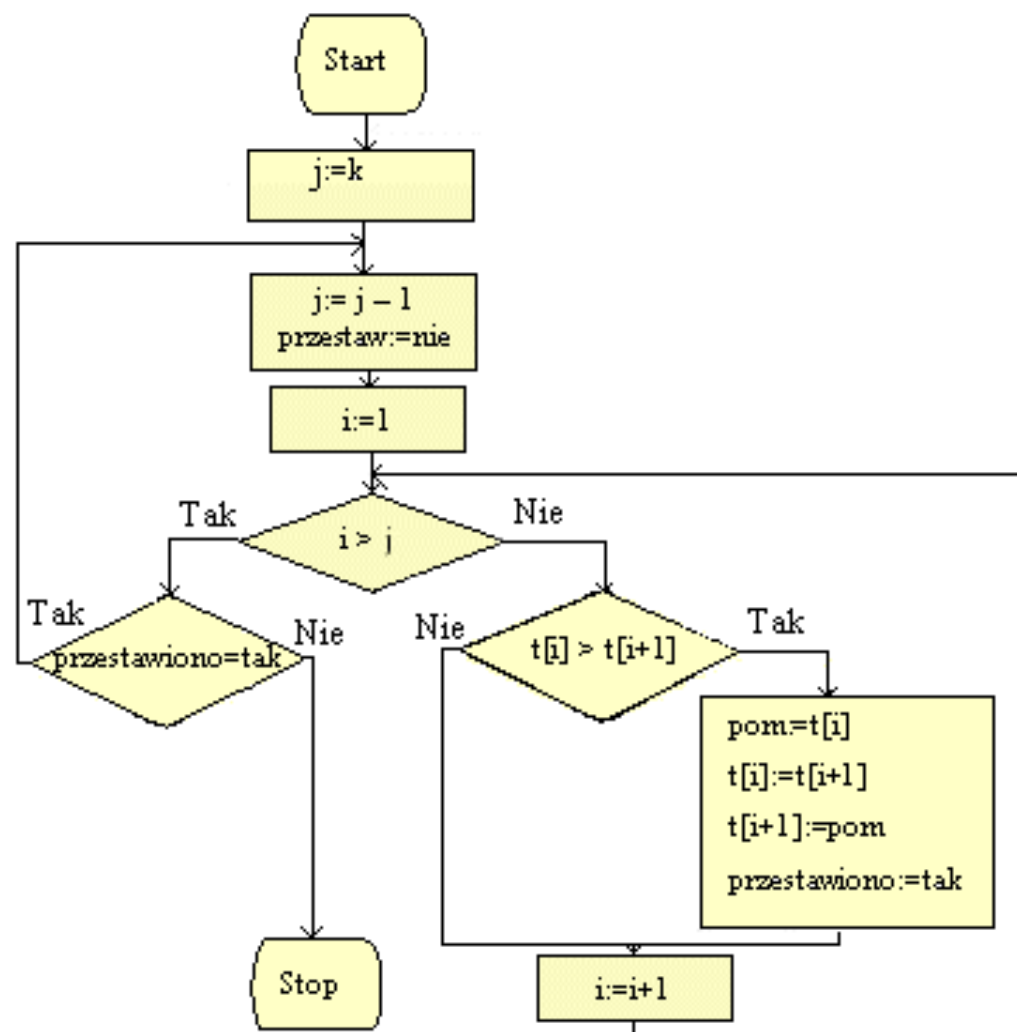
# Sortowanie bąbelkowe

## Lista kroków

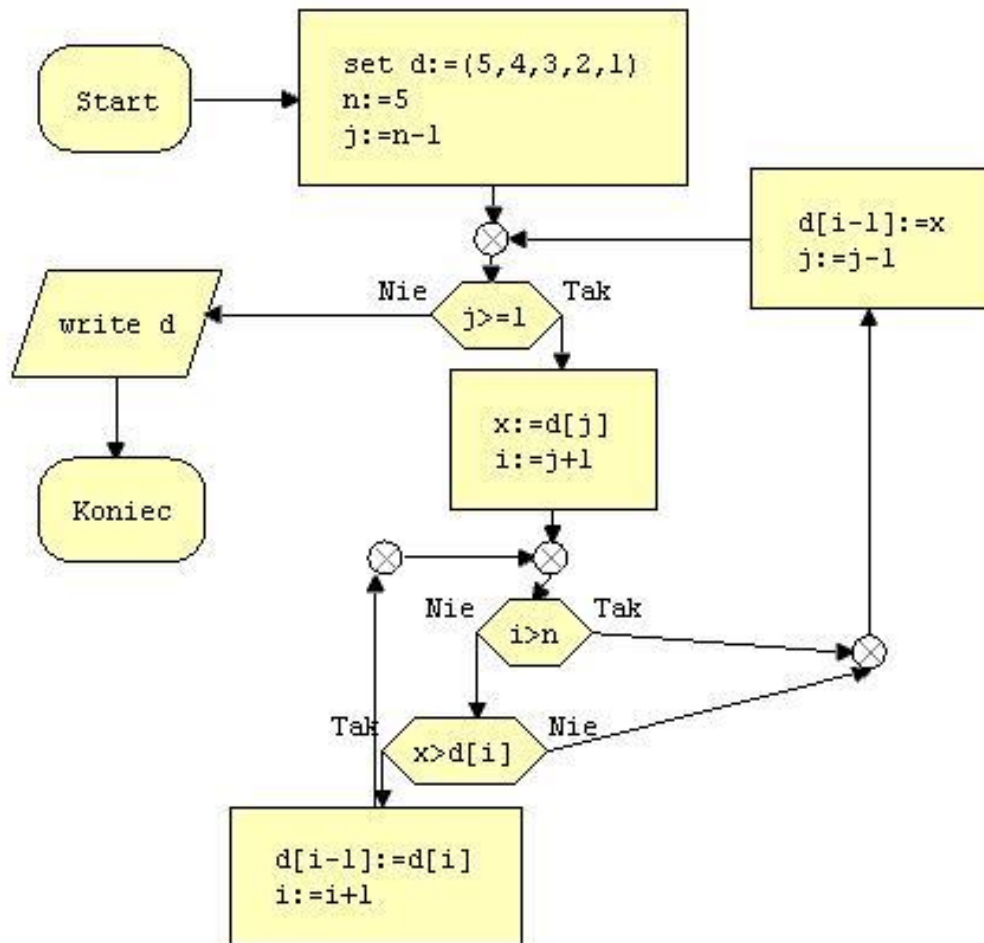
K01: Dla  $j = 1, 2, \dots, n - 1$ : wykonuj K02

K02: Dla  $i = 1, 2, \dots, n - 1$ : jeśli  $d[i] > d[i + 1]$ , to  $d[i] \leftrightarrow d[i + 1]$

K03: Zakończ



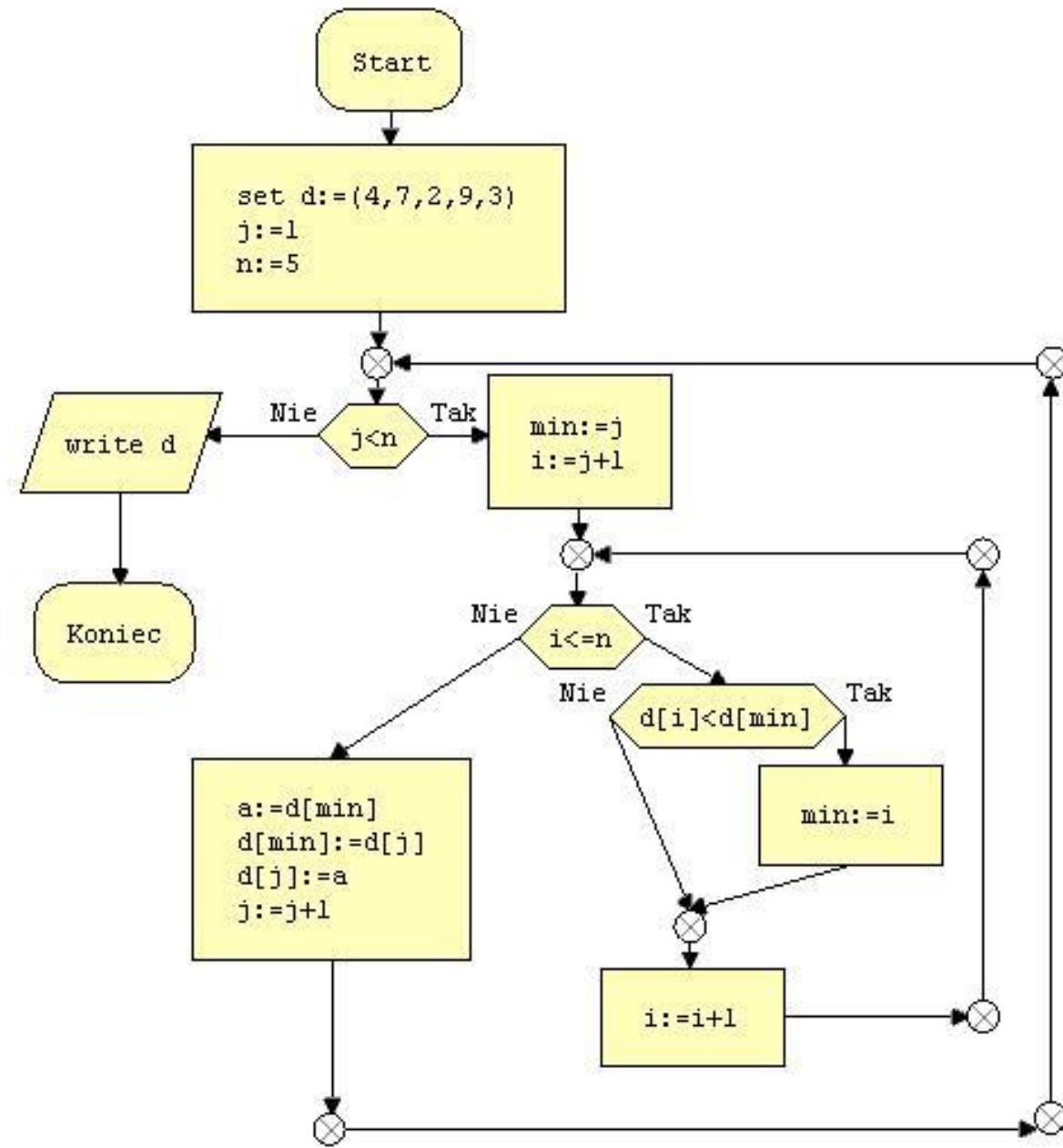
# Sortowanie przez wstawianie



## Lista kroków

- K01: Dla  $j = n - 1, n - 2, \dots, 1$ : wykonuj K02...K04
- K02:  $x \leftarrow d[j]; i \leftarrow j + 1$
- K03: Dopóki  $(i \leq n) \wedge (x > d[i])$ : wykonuj  $d[i - 1] \leftarrow d[i]; i \leftarrow i + 1$
- K04:  $d[i - 1] \leftarrow x$
- K05: Zakończ

# Sortowanie przez scalanie



## Lista kroków algorytmu scalania

K01:  $i_1 \leftarrow i_p$ ;  $i_2 \leftarrow i_s$ ;  $i \leftarrow i_p$

K02: Dla  $i = i_p, i_p + 1, \dots, i_k$ : wykonuj  
jeśli  $(i_1 = i_s) \vee (i_2 \leq i_k \wedge d[i_1] > d[i_2])$ , to  
 $p[i] \leftarrow d[i_2]$ ;  $i_2 \leftarrow i_2 + 1$   
inaczej  
 $p[i] \leftarrow d[i_1]$ ;  $i_1 \leftarrow i_1 + 1$

K03: Dla  $i = i_p, i_p + 1, \dots, i_k$ :  $d[i] \leftarrow p[i]$

K04: Zakończ