# Porównanie algorytmów sortujących

## Wstęp

Celem projektu było porównanie algorytmów sortujących o złożonościach: n2 i n(logn). Zostały wybrane następujące algorytmy:

1. Złożoność n2:
   1. Sortowanie bąbelkowe
   2. Sortowanie bąbelkowe dwukierunkowe
   3. Sortowanie przez wybór
2. Złożoność n(logn):
   1. Sortowanie przez kopcowanie
   2. Sortowanie przez scalanie
   3. Sortowanie szybkie

Program porównujący każdy z algorytmów został zrealizowany w JavaScript w środowisku uruchomieniowym Node.js (w wersji 21.7.3). Algorytmy sortowały trzy rodzaje tablic 1000-elementowych z zakresu <1, 1000>:

1. Tablicę liczb rosnących,
2. Tablicę liczb malejących,
3. Tablicę liczb losowych.

## Zestawienie wyników

Poniżej przedstawiam mój osobisty ranking algorytmów sortujących.

1. Quick sort – dobrze sobie radzi przy każdym zestawie liczb
2. Przez kopcowanie – zbliżony czas do Quick sort, jednak maksymalny czas sortowania zgodnego zestawu liczb jest 5,5 razy wolniejszy od Quick sort.
3. Przez wybór – niski średni czas sortowania
4. Przez scalanie – bardzo podobny czas dla każdego typu danych, niewielki maksymalny czas sortowania.
5. Bąbelkowe standardowe – długi czas trwania sortowania nawet posortowanych danych.
6. Bąbelkowe dwukierunkowe – największy czas trwania sortowania.

W załączniku przesyłam arkusz kalkulacyjny z dokładnymi wynikami uzyskanymi podczas badania.