

Data oddania sprawozdania: 13.03.2020

# SPRAWOZDANIE

## **Temat: Algorytmy sortujące.**

Wykonali: Kacper Chmielewski, Igor Charzewski

Poniższe tabele oraz wykresy prezentują czasy, które potrzebne są do wykonania zadania przez poszczególne algorytmy sortujące w zależności od ilości elementów do posortowania i rodzaju danych wejściowych.

Czas wykonania zadania dla losowych danych wejściowych.

Wielkość tablicy	Insertion Sort	Quick Sort	Selection Sort	Heap Sort
5	0:00:00	0:00:00	0:00:00	0:00:00
10	0:00:00	0:00:00	0:00:00	0:00:00
20	0:00:00	0:00:00	0:00:00	0:00:00
40	0:00:00	0:00:00	0:00:00	0:00:00
80	0:00:00	0:00:00	0:00:00	0:00:00
160	0:00:00	0:00:00.00100 1	0:00:00	0:00:00.00100 1
320	0:00:00.00100 1	0:00:00.00400 4	0:00:00.00400 3	0:00:00.00100 1
640	0:00:00	0:00:00.01401 2	0:00:00.01601 5	0:00:00.00300 2
1280	0:00:00	0:00:00.05104 6	0:00:00.06706 1	0:00:00.00600 5
2560	0:00:00	0:00:00.12411 2	0:00:00.24822 4	0:00:00.01301 1

Czas wykonania zadania dla rosnących danych wejściowych.

Wielkość tablicy	Insertion Sort	Quick Sort	Selection Sort	Heap Sort
5	0:00:00	0:00:00	0:00:00	0:00:00
10	0:00:00	0:00:00	0:00:00	0:00:00
20	0:00:00	0:00:00.00100 1	0:00:00.00100 0	0:00:00
40	0:00:00	0:00:00	0:00:00	0:00:00
80	0:00:00	0:00:00	0:00:00	0:00:00
160	0:00:00	0:00:00.00100 0	0:00:00.00100 2	0:00:00.00100 1
320	0:00:00	0:00:00.00400 4	0:00:00.00400 3	0:00:00.00100 1
640	0:00:00	0:00:00.01401 3	0:00:00.01601 4	0:00:00.00300 3
1280	0:00:00.00100 1	0:00:00.04904 4	0:00:00.06405 8	0:00:00.00600 5
2560	0:00:00.00100 1	0:00:00.12411 1	0:00:00.25723 1	0:00:00.01301 2

Czas wykonania zadania dla malejących danych wejściowych.

Wielkość tablicy	Insertion Sort	Quick Sort	Selection Sort	Heap Sort
5	0:00:00	0:00:00	0:00:00	0:00:00
10	0:00:00	0:00:00	0:00:00	0:00:00
20	0:00:00	0:00:00	0:00:00	0:00:00
40	0:00:00	0:00:00	0:00:00	0:00:00
80	0:00:00	0:00:00.001001	0:00:00	0:00:00
160	0:00:00	0:00:00.001000	0:00:00	0:00:00.001001
320	0:00:00	0:00:00.004004	0:00:00.003003	0:00:00.001001
640	0:00:00.001001	0:00:00.015014	0:00:00.016014	0:00:00.003003
1280	0:00:00	0:00:00.050045	0:00:00.065066	0:00:00.006005
2560	0:00:00	0:00:00.127114	0:00:00.253220	0:00:00.014013

Czas wykonania zadania dla V-kształtnych danych wejściowych.

Wielkość tablicy	Insertion Sort	Quick Sort	Selection Sort	Heap Sort
5	0:00:00	0:00:00	0:00:00	0:00:00
10	0:00:00	0:00:00	0:00:00	0:00:00
20	0:00:00	0:00:00	0:00:00	0:00:00
40	0:00:00	0:00:00	0:00:00	0:00:00
80	0:00:00	0:00:00	0:00:00	0:00:00.001001
160	0:00:00	0:00:00.001001	0:00:00.001001	0:00:00
320	0:00:00	0:00:00.003002	0:00:00.004003	0:00:00.001000
640	0:00:00	0:00:00.014013	0:00:00.015014	0:00:00.002002
1280	0:00:00	0:00:00.051046	0:00:00.064058	0:00:00.005004
2560	0:00:00.001001	0:00:00.122110	0:00:00.255230	0:00:00.013012

## Złożoność algorytmów

Nazwa	Przypadek najlepszy	Przypadek losowy	Przypadek najgorszy
Quick Sort	$\Omega(n \log(n))$	$O(n \log(n))$	$O(n^2)$
Insertion Sort	$\Omega(n)$	$O(n^2)$	$O(n^2)$
Selection Sort	$\Omega(n^2)$	$O(n^2)$	$O(n^2)$
Heap Sort	$\Omega(n \log(n))$	$O(n \log(n))$	$O(n \log(n))$

Z wyników tabel umieszczonych w dokumencie możemy zauważyć, że najbardziej wydajnym algorytmem sortującym jest Heap sort. Jego czas pracy rośnie logarymicznie w zależności od rozmiaru instancji, dodatkowo wpływ przypadku jest znikomy w porównaniu do innych algorytmów.

Najgorszym(najwolniejszym) algorytmem jest natomiast Selection sort. Wynika to z faktu, iż jego złożoność wynosi zawsze  $n^2$  niezależnie od trudności przypadku.

Czas pracy Selection Sort jest spowodowany tym, że operacje są wykonywane dla wszystkich indeksów sortowanej tablicy.