

# Analiza statystyczna algorytmów

testy zostały posortowane 10 razy dla tablic:

małych – 800 elementów

Duże- 10 000 elementów (ze względu na słaby procesor nie zastosowano zalecanej ilości 100 000)

Bubblesort :

Dla małych:

srednia dla malej tablicy 0.10202057

odchylenie standardowe dla malej tablicy 0.0015580470000000097

Dla dużych:

srednia dla duzej tablicy 16.22702465

odchylenie standardowe dla duzej tablicy 0.05554825499999971

InsertionSort:

Dla małych:

srednia dla malej tablicy 0.076625980000000001

odchylenie standardowe dla malej tablicy 0.0025181720000000047

Dla dużych:

srednia dla duzej tablicy 12.78690645

odchylenie standardowe dla duzej tablicy 0.11988595499999981

heapSort:

srednia dla malej tablicy 0.007814699999999994

odchylenie standardowe dla malej tablicy za małe ułamki do obliczenia przez  
pythona

srednia dla duzej tablicy 0.160283059999999998

odchylenie standardowe dla duzej tablicy 0.00280725600000000192

quicksort:

srednia dla malej tablicy 0.0043735900000000002

odchylenie standardowe dla malej tablicy za małe ułamki do obliczenia przez  
pythona

srednia dla duzej tablicy 0.087574520000000003

odchylenie standardowe dla duzej tablicy 0.00145585199999999838

mergesort:

srednia dla malej tablicy 0.0055503200000000003

odchylenie standardowe dla malej tablicy 0.000141812

srednia dla duzej tablicy 0.095137290000000001

odchylenie standardowe dla duzej tablicy 0.00103639099999999948

wnioski: dla danych losowych nie zależnie od wielkości tablicy najlepszym wyborem będzie:

1. Quick sort
2. Merge sort
3. Heap sort
4. Insertion sort
5. Bubble sort

Warto dodać że gdyby dane były już wstępnie posortowane wynik sortowania quick sort był by gorszy