**Ilość danych : 100 000 ( sortowanie rosnące)**

Pomiary czasowe dla wybranych algorytmów sortowania

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | InsertSort | BubbleSort | SelectSort | QuickSort | HeapSort | MergeSort |
| Dane losowe | 3221ms | 17873ms | 2887ms | 12ms | 16ms | 12ms |
| Dane posortowane rosnąco | 1ms | 1ms | 1789ms | 13ms | 15ms | 7ms |
| Dane posortowane malejąco | 6517ms | 9432ms | 2961ms | 9ms | 15ms | 12ms |

WNIOSKI

InsertSort, BubbleSort oraz MergeSort są sortowniami stabilnymi, natomiast QuickSort, SelectSort oraz HeapSort niestabilnymi.

Przy wyborze algorytmu sortowania powinniśmy się zastanowić jakie i ile danych będziemy sortować. W przypadku małej ilości możemy wybrać InsertSort, BubbleSort lub SelectSort, bo są łatwe w implementacji oraz przy takiej ilości danych różnice pomiędzy wszystkimi algorytmami są nieznaczące.

Problem zaczyna się, gdy naszych danych jest więcej niż 5000, wtedy gdy nie wiemy czy dane są wstępnie posortowane najlepiej skorzystać z MergeSort’a lub QuickSorta’a, a w przypadku gdy dane w wieksząości są posortowane rosnąco powinniśmy rozważyć implementację BubbleSort’a lub InsertSort’a