

# SEBASTIAN NOWAK

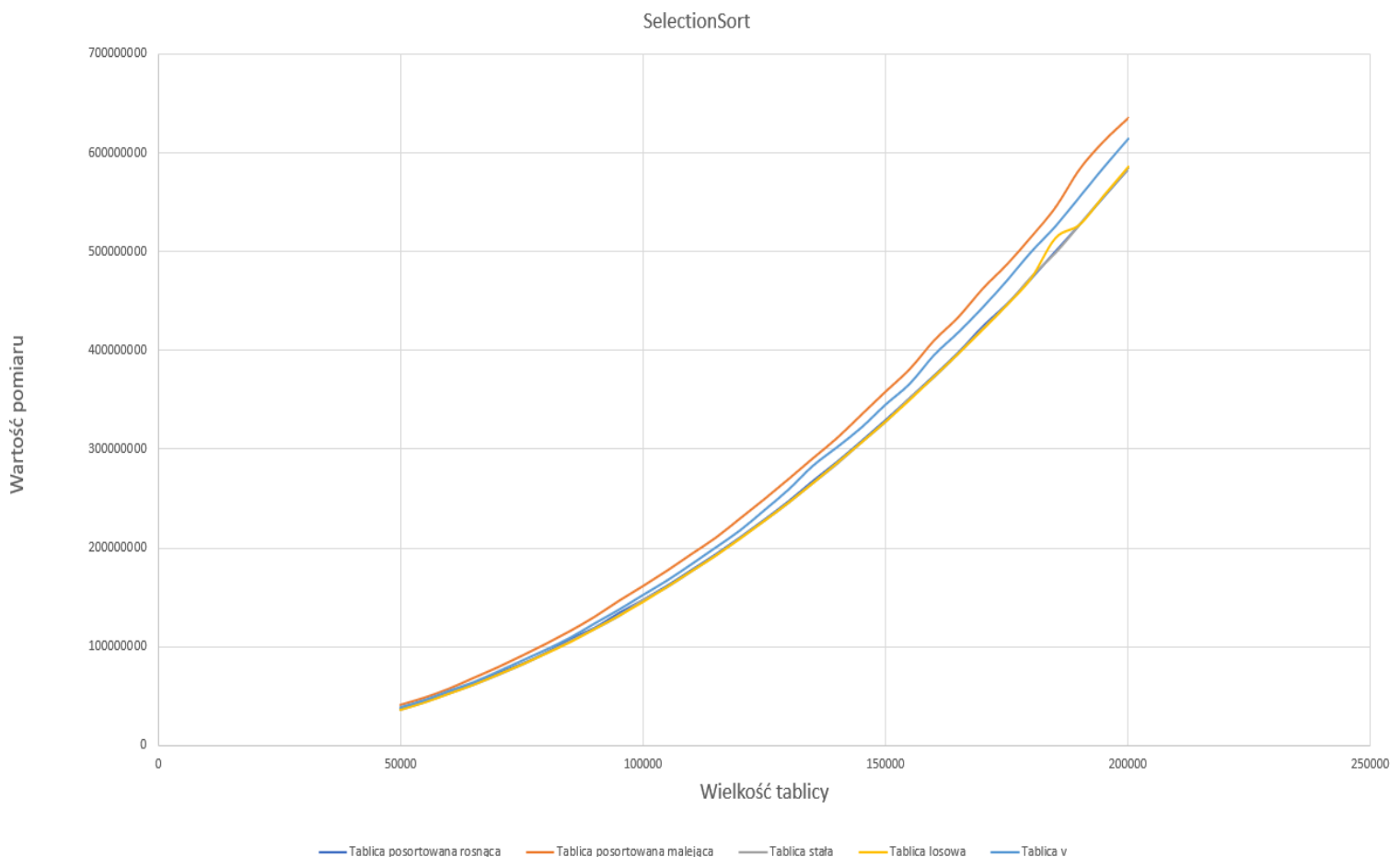
## Sprawozdanie do projektu 3

### Zbadanie algorytmów sortujących

Projekt polegał na zbadaniu algorytmów sortujących (SelectionSort, InsertionSort, CocktailSort, HeapSort, QuickSort) z różnymi danymi wejściowymi (tablicami posortowanych (ściśle) rosnąco, posortowanych (ściśle) malejąco), stałych, losowych, V-kształtnych, A-kształtnych).

#### 1.

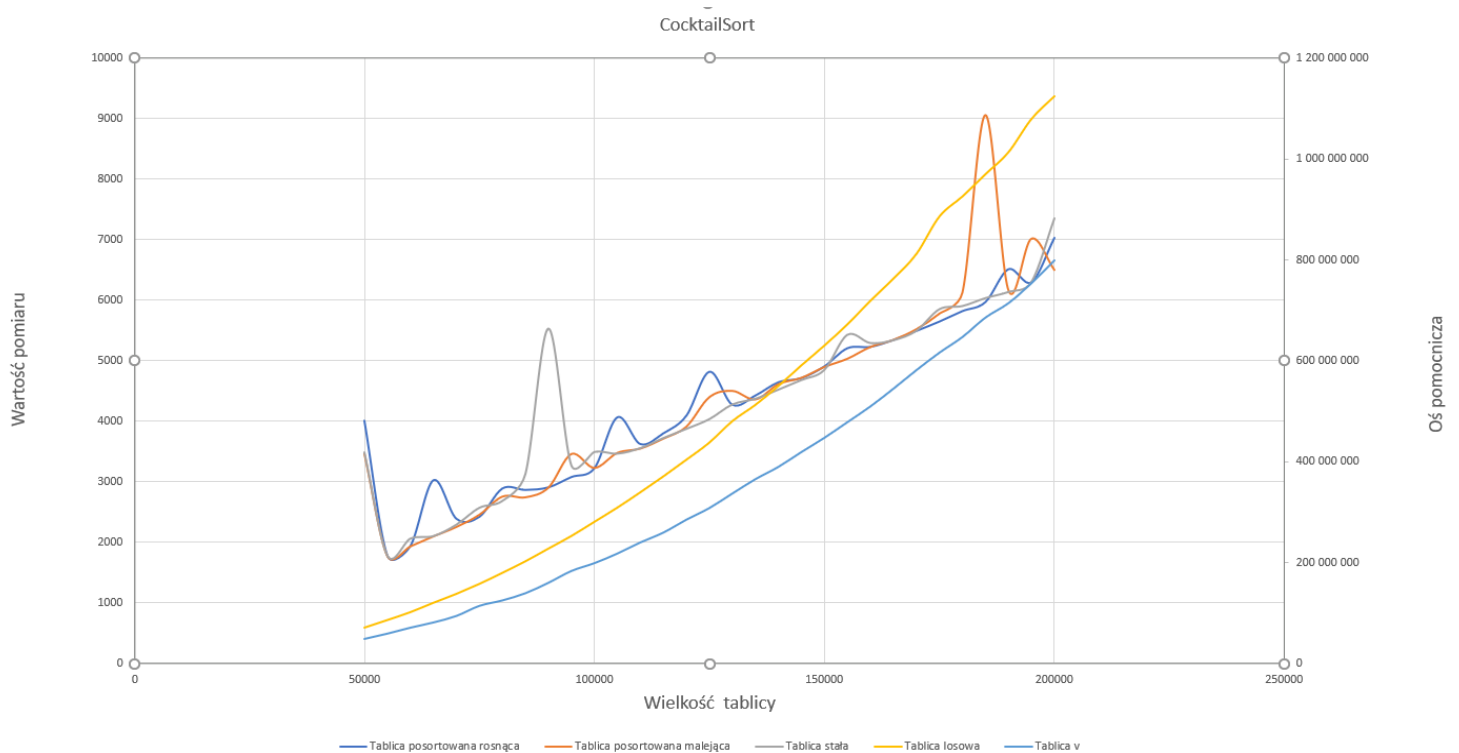
Zbadanie każdego z algorytmów z różnymi danymi wejściowymi



Różnice w czasie wykonania przy wprowadzeniu różnych serii danych jest niewielka. Najgorzej algorytm poradził sobie z tablicą posortowaną malejąco, a najlepiej z tablicą posortowaną rosnąco.



W wykresie dla serii danych z tablicy losowej i v-kształtnej została dodana oś pomocnicza z nimi też algorytm poradził sobie najgorzej. Pozostałe tablice wypadły bardzo podobnie do siebie.



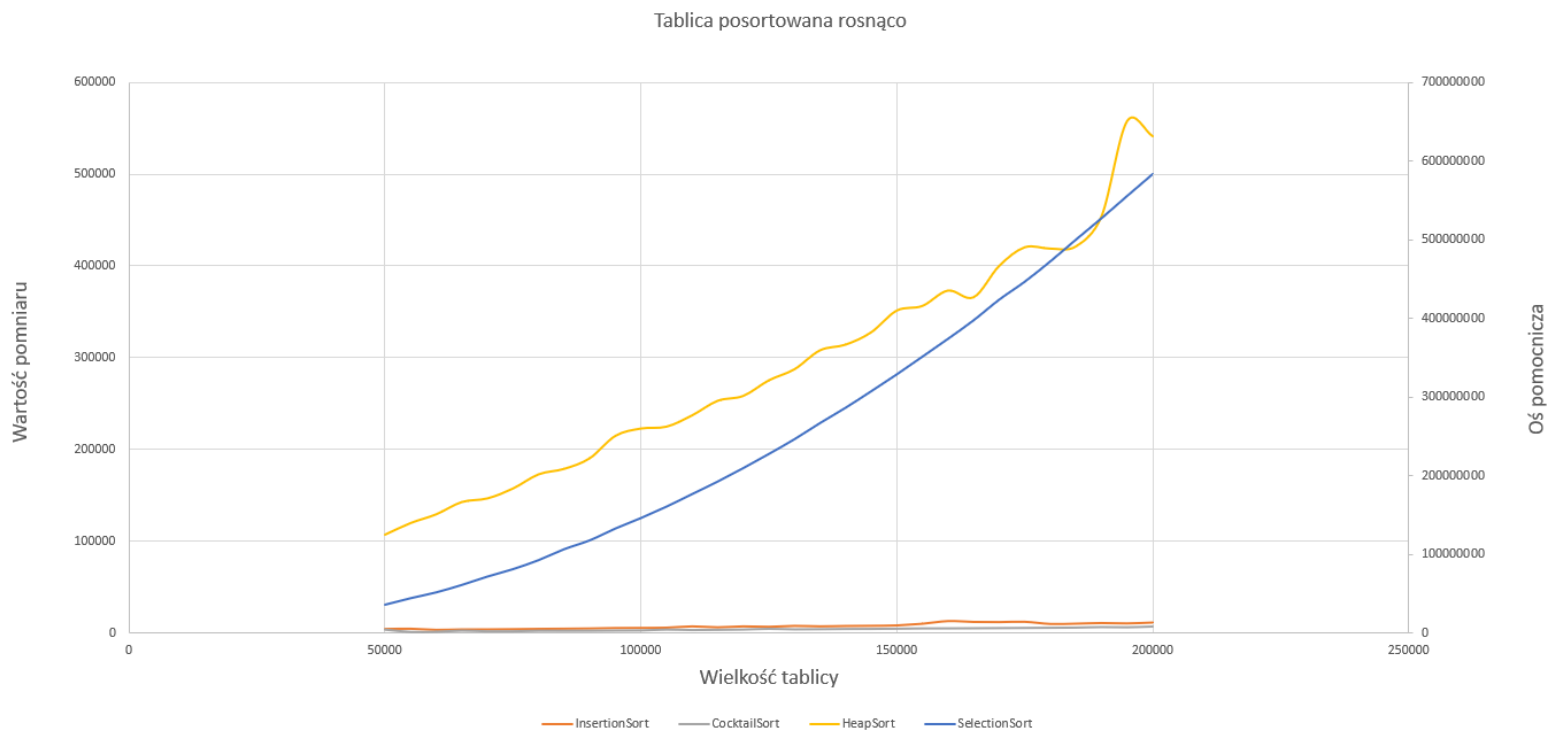
W wykresie dla serii danych z tablicy losowej i v-kształtnej została dodana oś pomocnicza z nimi też algorytm poradził sobie najgorzej. Pozostałe tablice wypadły bardzo podobnie do siebie.



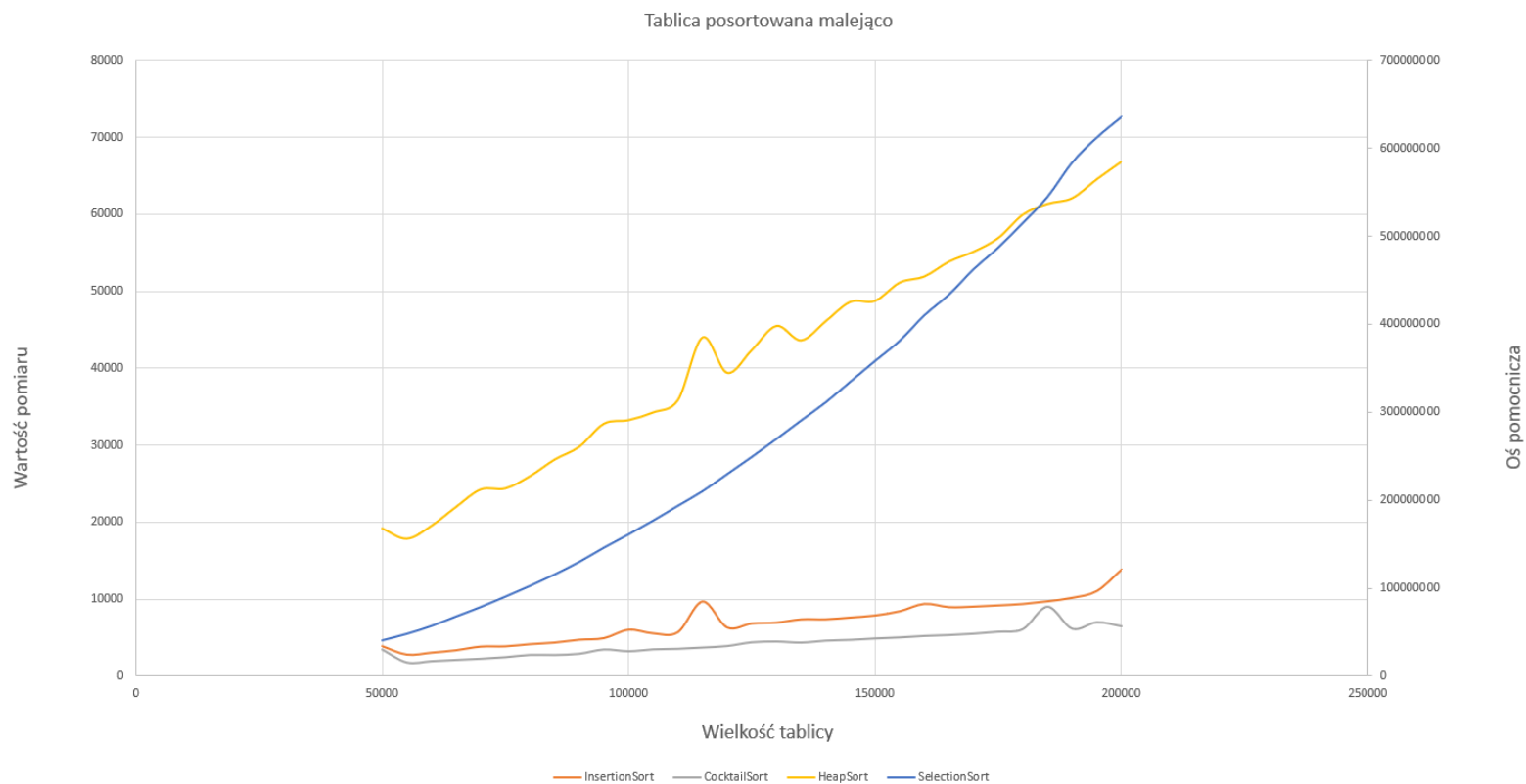
HeapSort najlepiej poradził sobie z tablicą malejącą i stałą, a najgorzej z tablicą losową.

## Część 2.

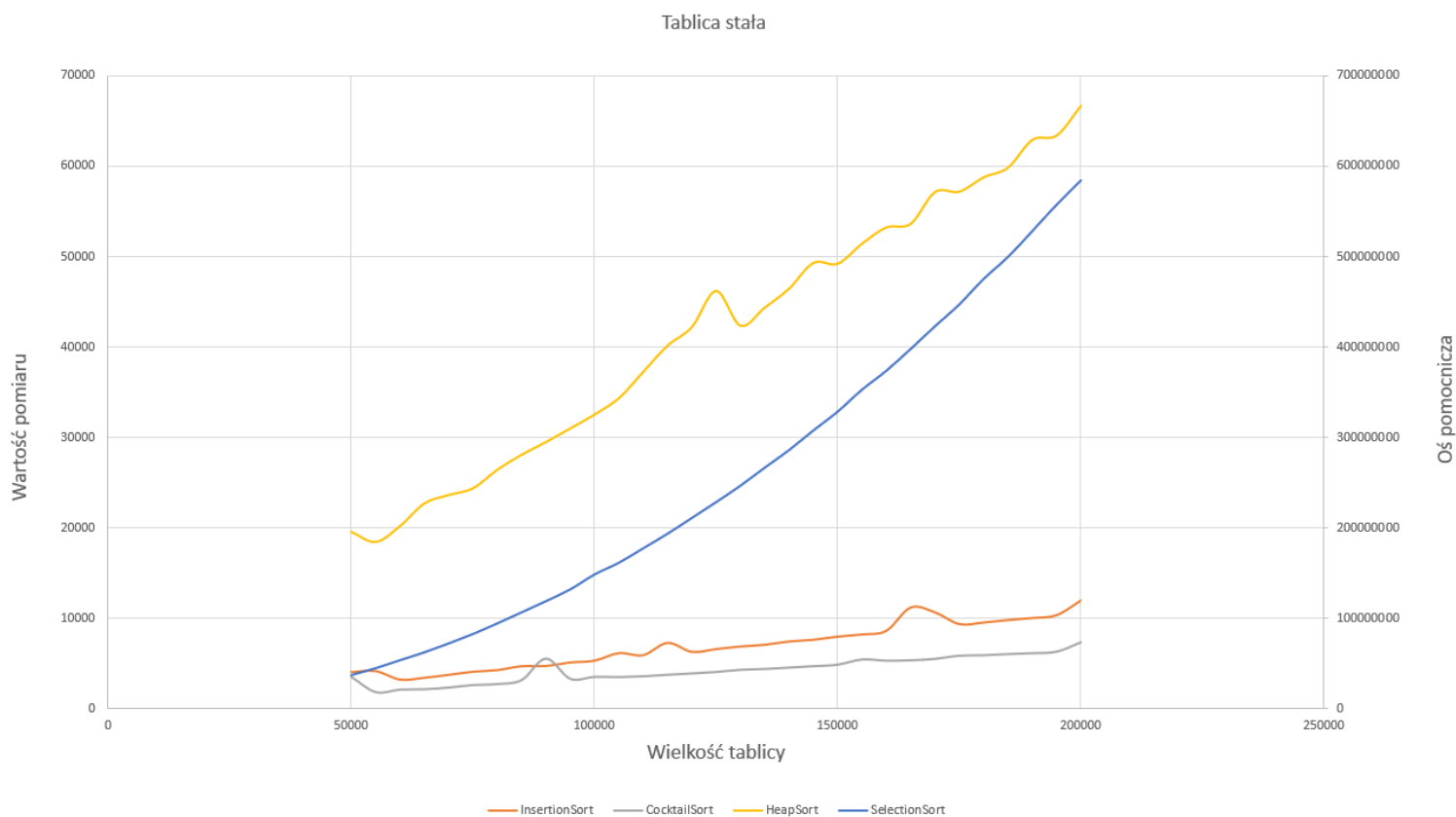
Sprawdzenie jaki typ danych jest najlepszy i najgorszy dla danego typu danych wejściowych.



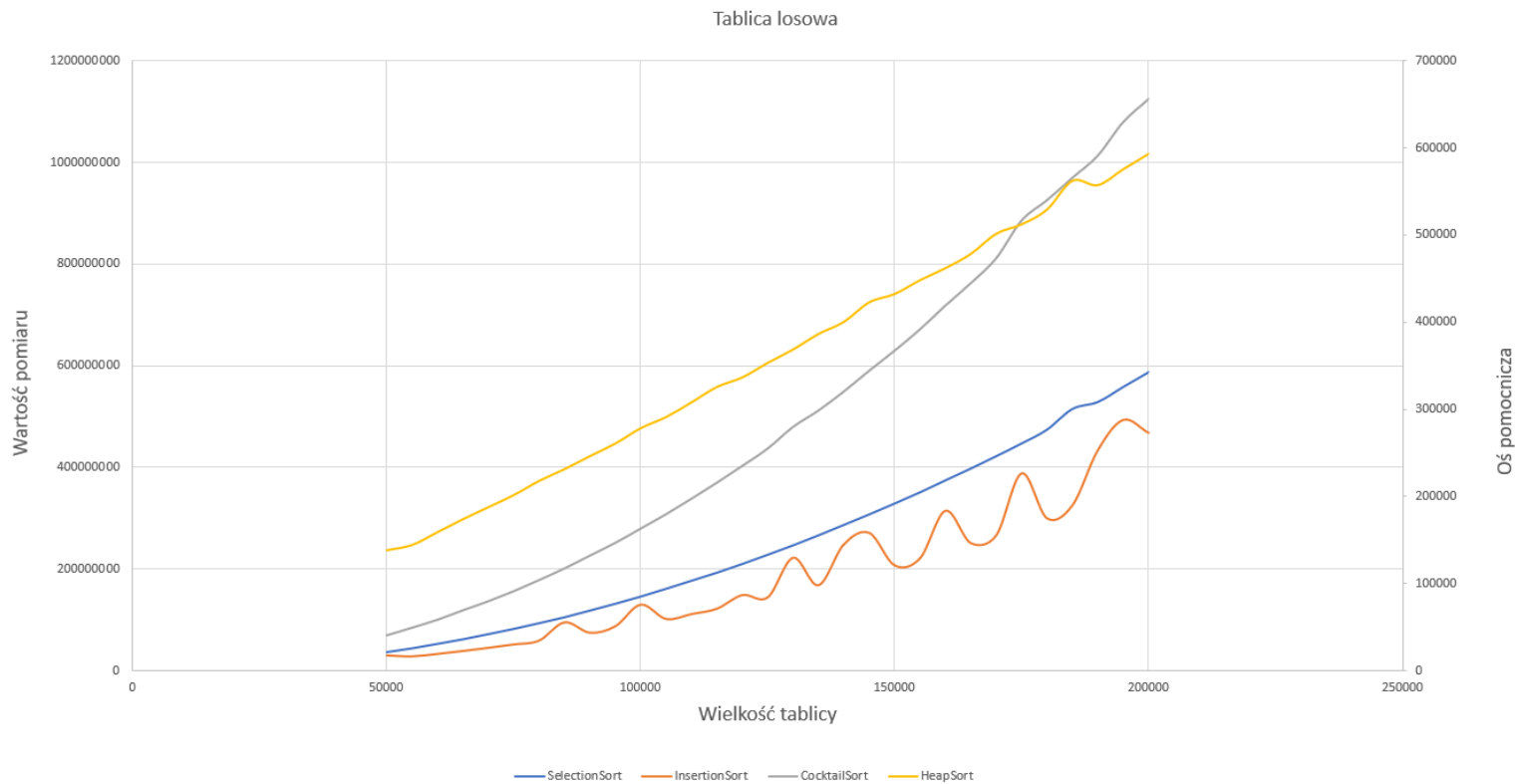
Dla tablicy posortowanej rosnąco najlepiej poradził sobie CocktailSort i InsertionSort. Najgorzej poradził sobie SelectionSort dla niego też została dodana oś pomocnicza.



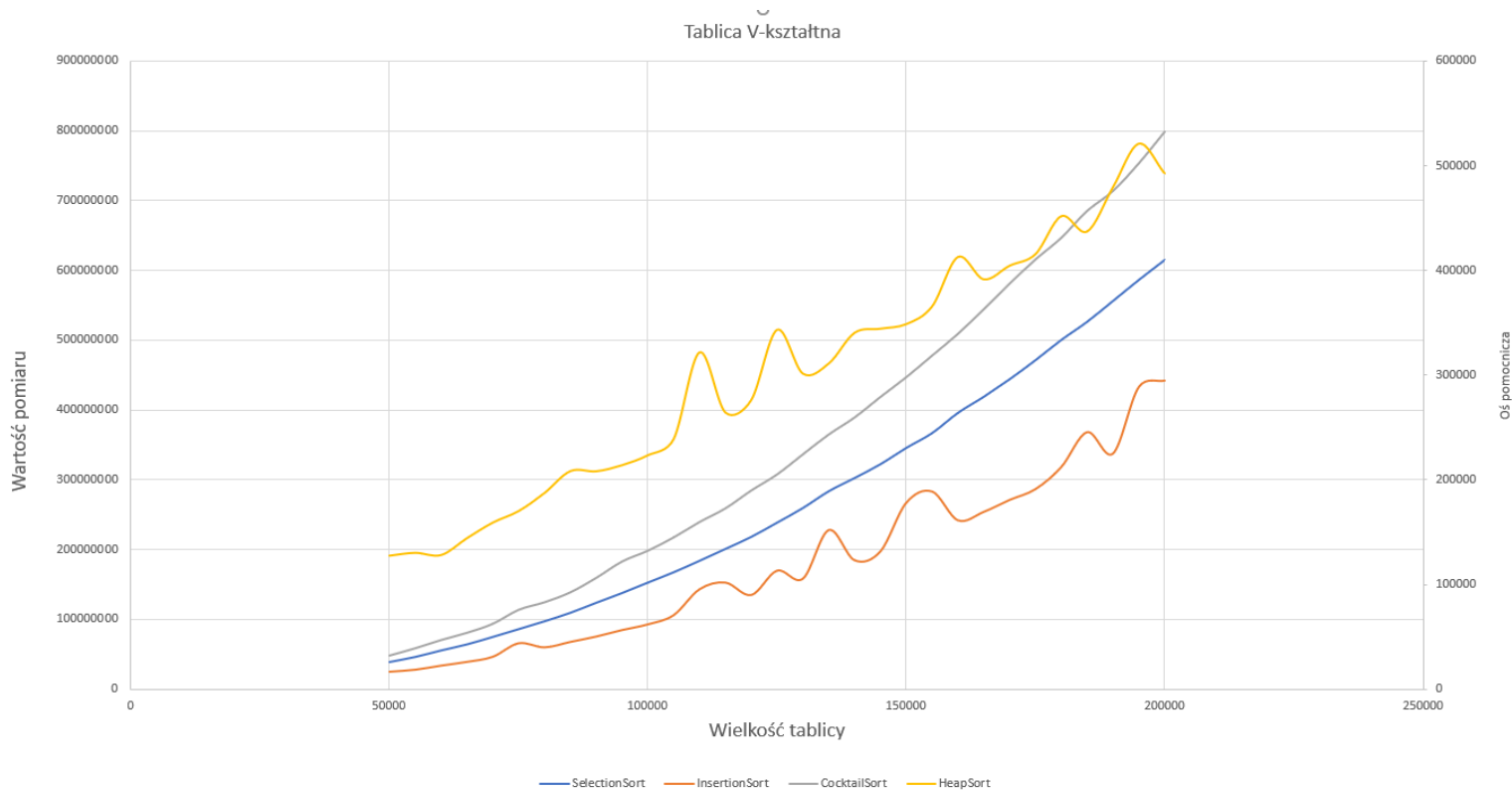
Dla tablicy posortowanej malejąco najlepiej poradził sobie CocktailSort. Najgorzej poradził sobie SelectionSort dla niego też została dodana oś pomocnicza.



Dla tablicy stałej najlepiej poradził sobie CocktailSort. Najgorzej poradził sobie SelectionSort dla niego też została dodana oś pomocnicza.



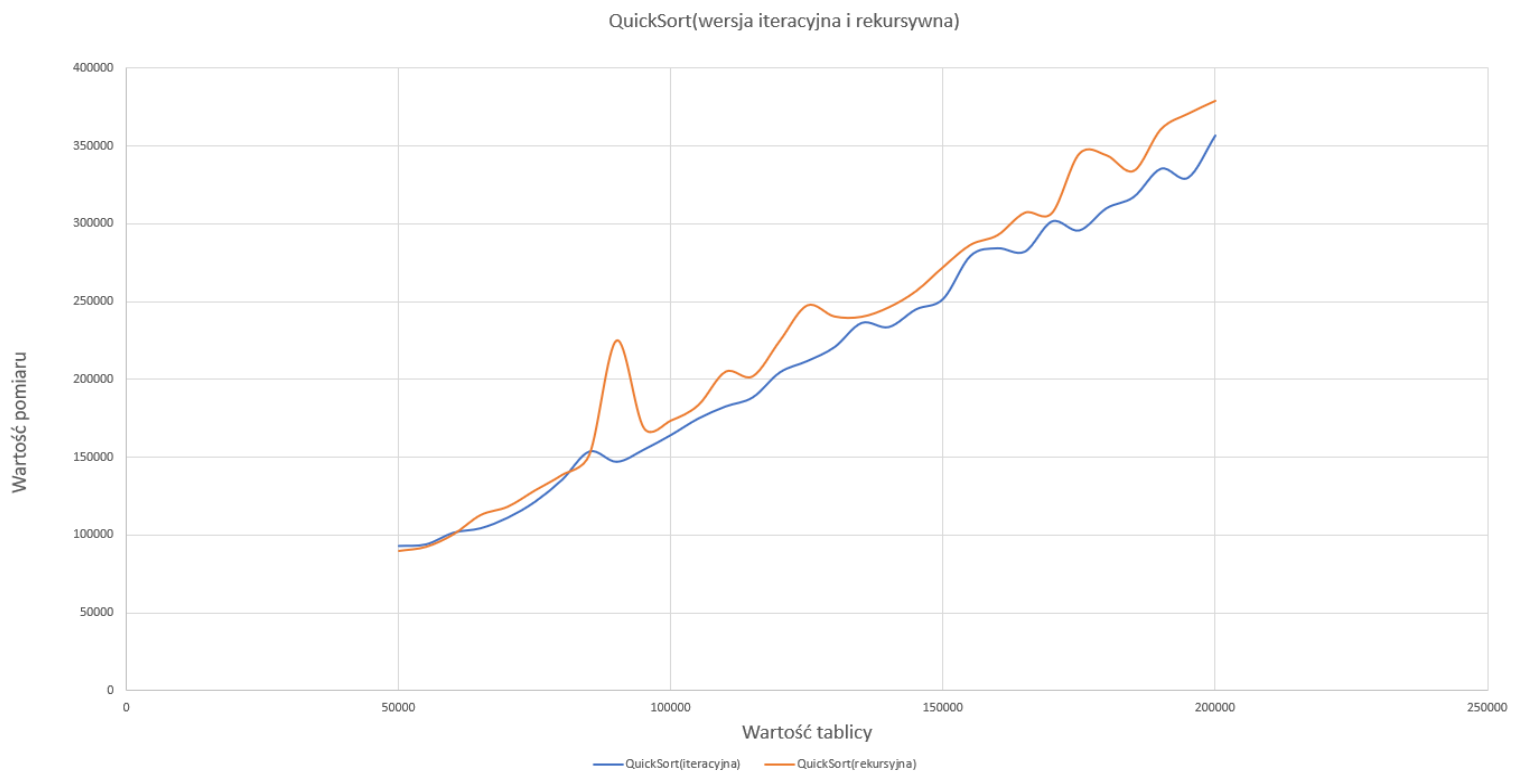
Dla tablicy losowej najlepiej poradził sobie HeapSort dla niego też została dodana oś pomocnicza. Najgorzej poradził sobie CocktailSort.



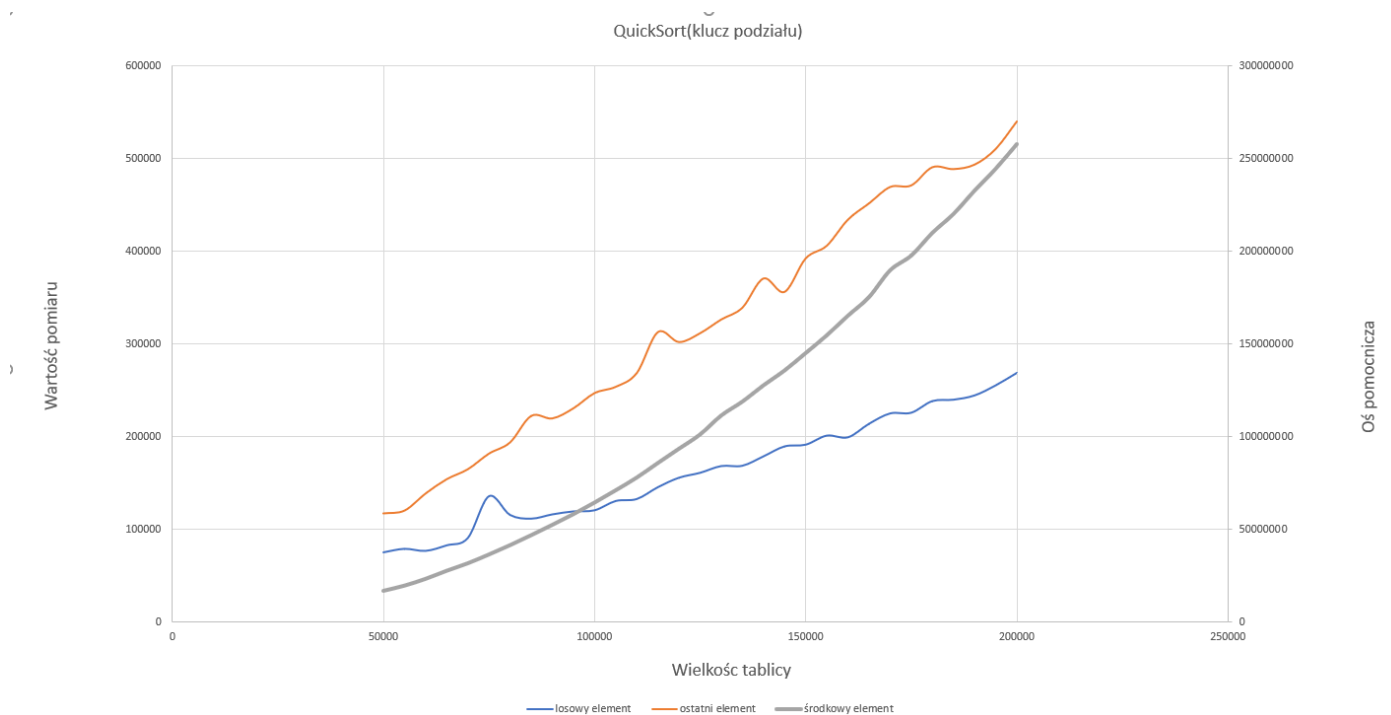
Dla tablicy losowej najlepiej poradził sobie HeapSort dla niego też została dodana oś pomocnicza. Najgorzej poradził sobie CocktailSort.

## Część 3.

Porównanie algorytmu QuickSort w wersji iteracyjnej i rekursywnej oraz ze względu na wybór klucza podziału (pivot).



Dla tablicy która zawiera liczby losowe lepszy okazał się QuickSort w wersji iteracyjnej.



Dla tablicy A-kształtnej najszybszy był QuickSort z kluczem podziału losowym. Najgorsza jest wersja z środkowym kluczem podziału dla niego została dodana oś pomocnicza (QuickSort w wersji iteracyjnej).