

# Projekt 3

## Algorytmy i struktury danych

Wykonał: Wojciech Nowakowski K35.2

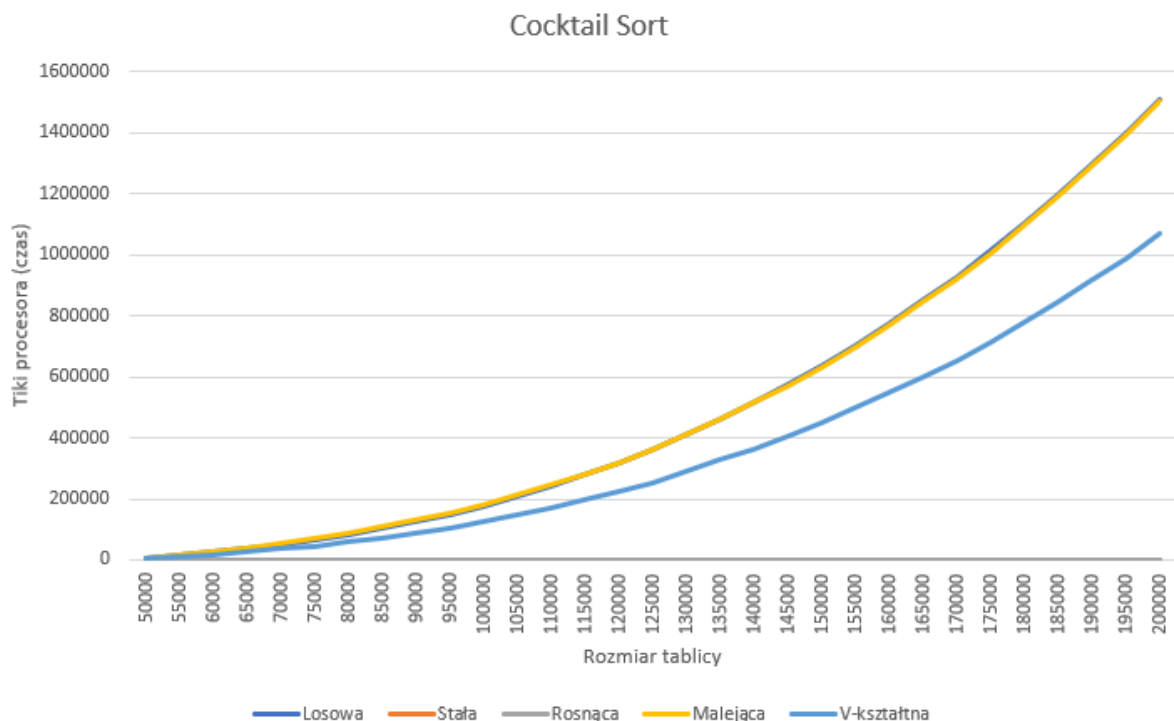
Celem projektu było sprawdzenie efektywności czterech algorytmów sortujących, poprzez zmierzenie czasu, w którym się wykona.

Obiektem badań będą kolejno: SelectionSort, InsertionSort, CocktailSort, oraz HeapSort.

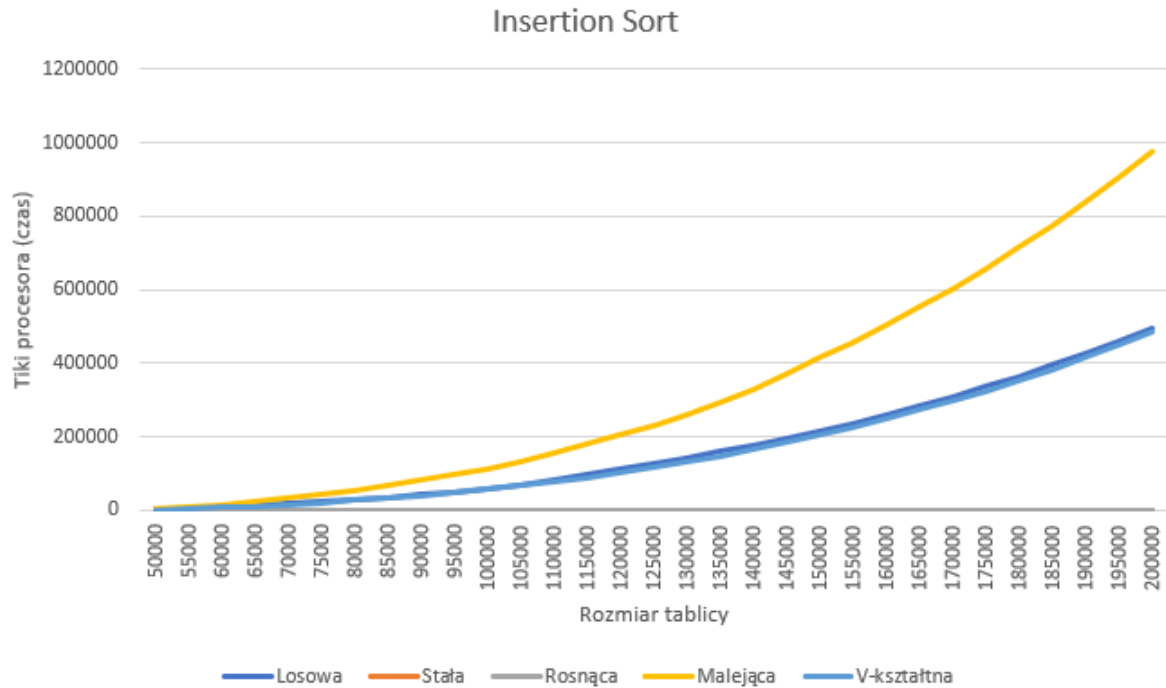
Punktów pomiarowych jest 31, zaczynają się od 5000 i wzrastają o 5 tysięcy, aż do 200000.

Pomiary wykonywane będą na pięciu rodzajach tablic: posortowanych rosnąco, posortowanych malejąco, stałych, losowych, oraz V-kształtnych.

## Część 1

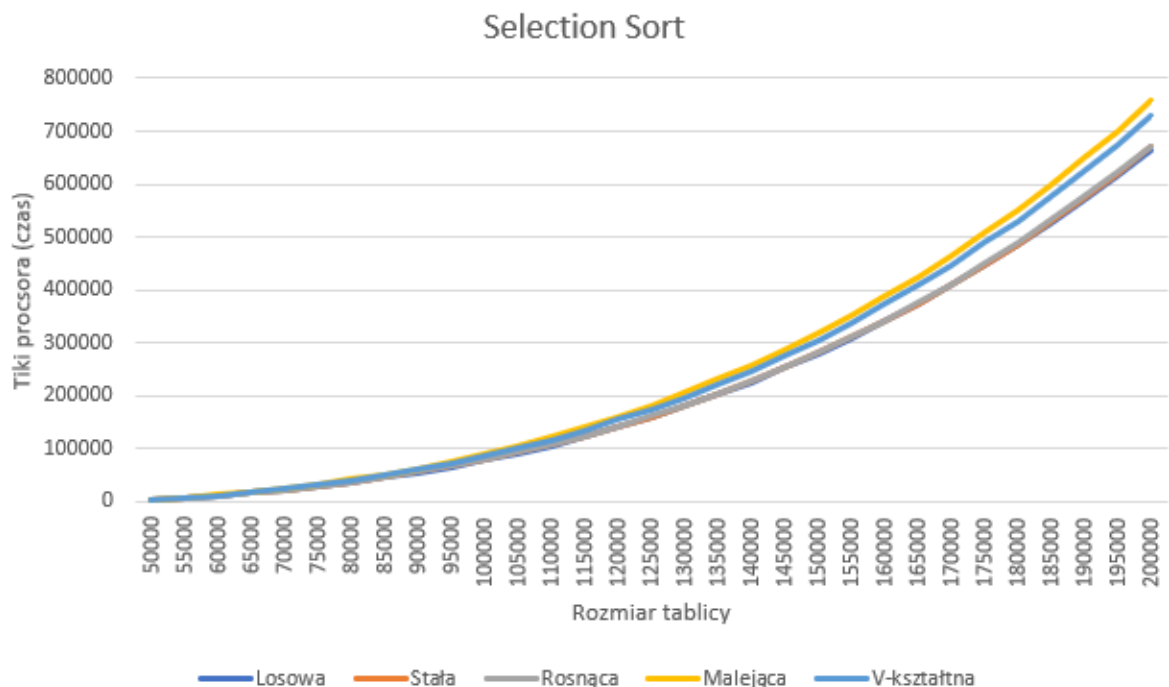


Przez zbliżone wyniki, oraz wybraną skalę, dane z tablicy stałej i rosnącej zlewają się w jedną całość, oraz wyglądają jak funkcja liniowa, mimo że nią nie są. W przypadku sortowania koktajlowego algorytm działa najlepiej na tablicy stałej, a najgorzej na malejącej.

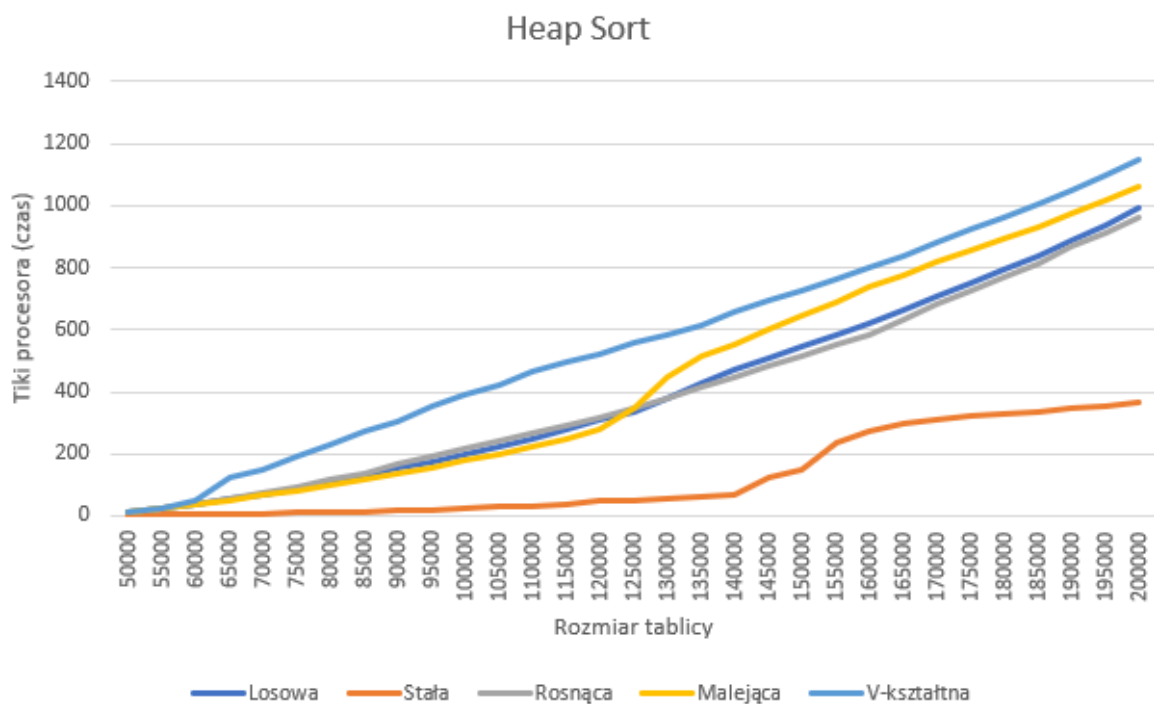


Przez zbliżone wyniki, oraz wybraną skalę, dane z tablicy stałej i rosnącej zlewają się w jedną całość, oraz wyglądają jak funkcja liniowa, mimo że nią nie są.

Czas działania Insertion Sort wygląda podobnie, jak w poprzednim przypadku, działa najlepiej z tablicami stałymi, a najgorzej z malejącymi.

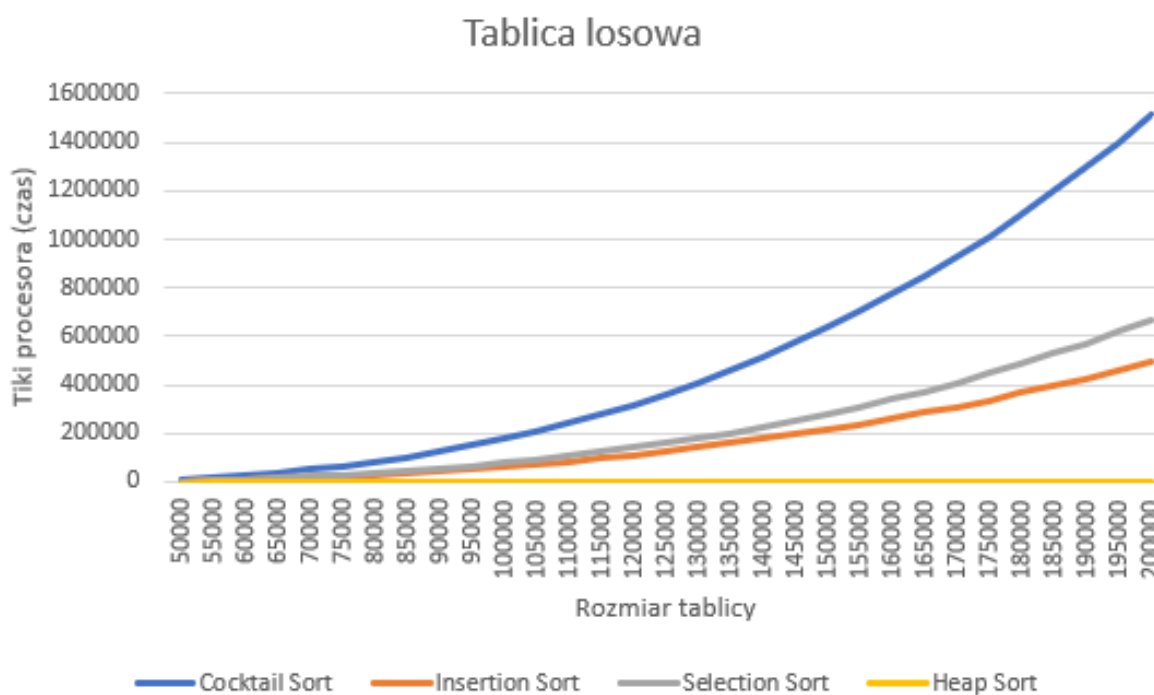


Tablica losowa, stała, oraz rosnąca, ze względu na podobne dane, oraz użytą skalę zlewają się w jedną całość. Selection Sort działa najszybciej z tablicą losową, a najgorzej z malejącą.

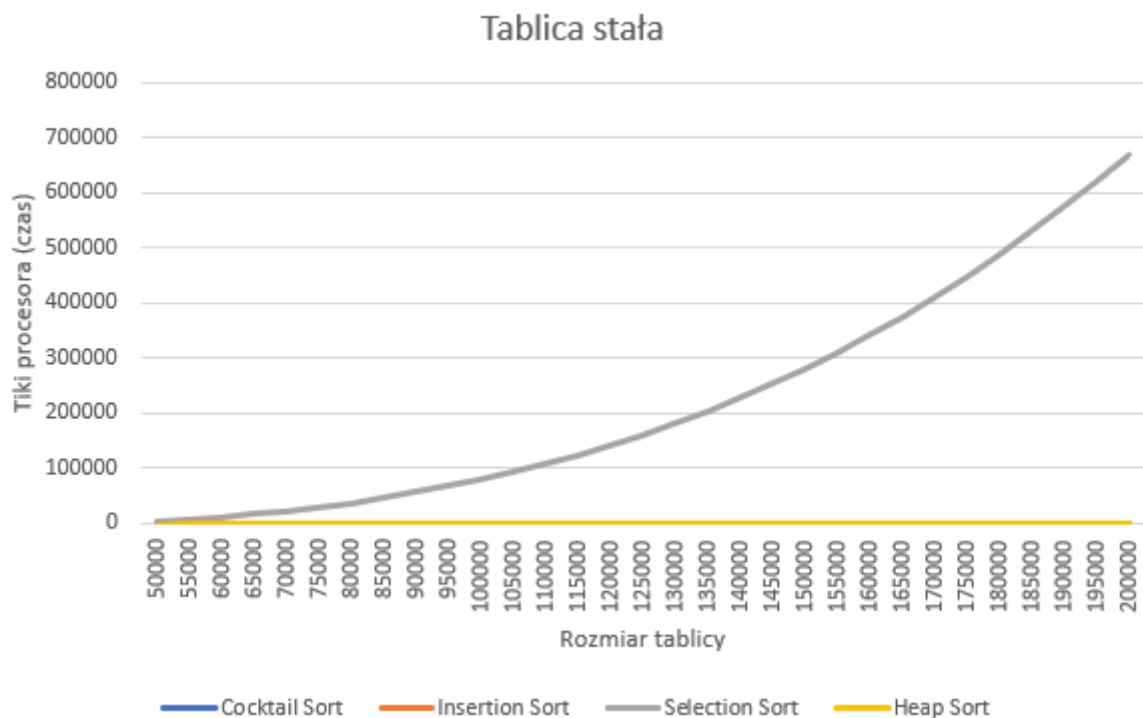


Algorytm Heap Sort zadziałał najszybciej dla tablicy stałej, a najgorzej dla V-kształtnej.

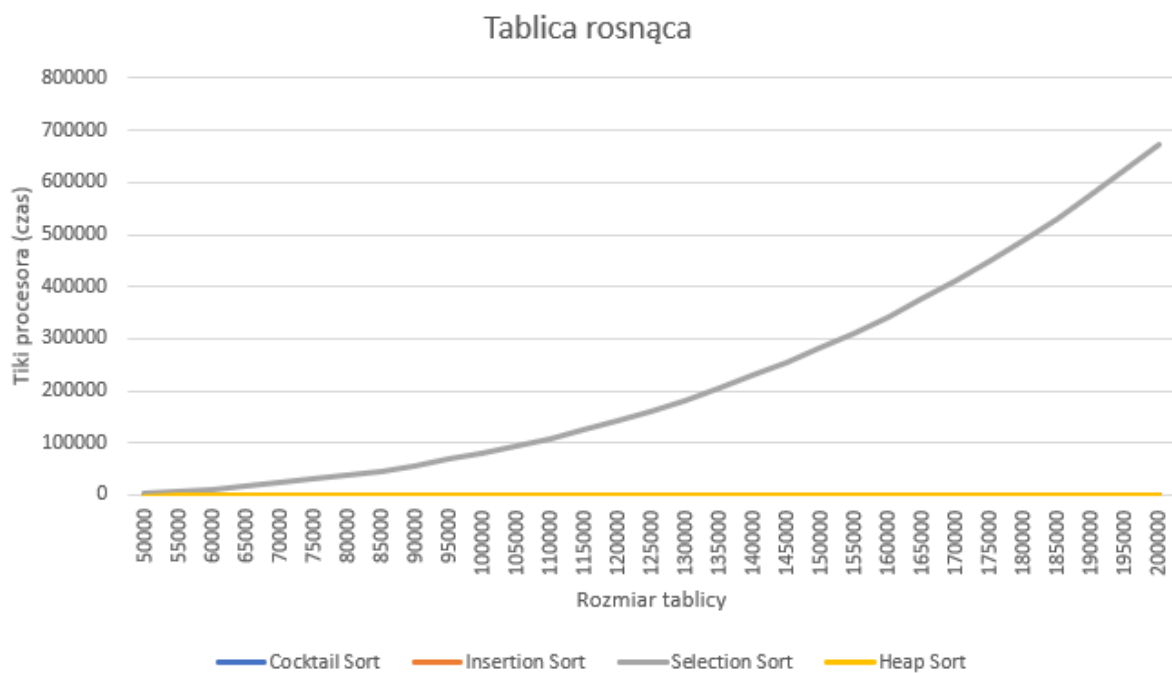
## Część 2



W wyniku eksperymentu możemy określić, że w przypadku tablic losowych najlepiej spisze się Heap Sort, z kolei najgorszy będzie Coctail Sort. Heap Sort z powodu obranej skali wygląda na funkcję liniową, jednak w rzeczywistości nią nie jest.

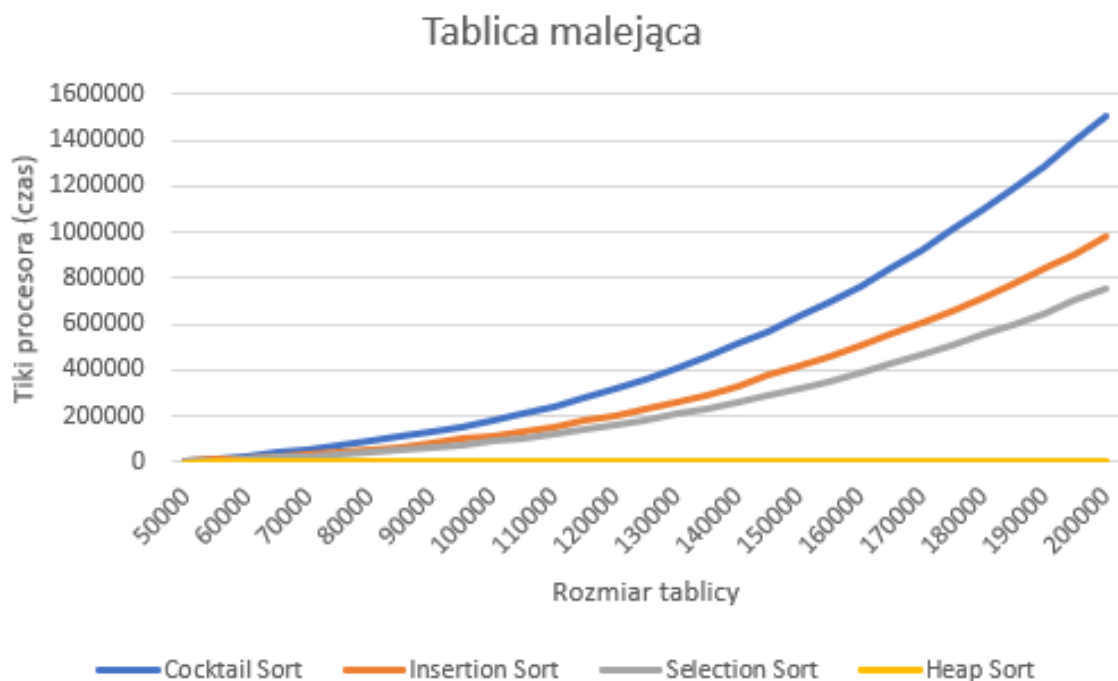


Ze względu na obraną skalę, oraz zbliżone wyniki, dane dotyczące Cocktail Sort, Insertion Sort, oraz Heap Sort zlewają się w jedną całość. W przypadku tablic stałych najskuteczniejsze okazał się Cocktail Sort, a najgorszy dla Selection Sort. Heap Sort z powodu obranej skali wygląda na funkcję liniową, jednak w rzeczywistości nią nie jest.

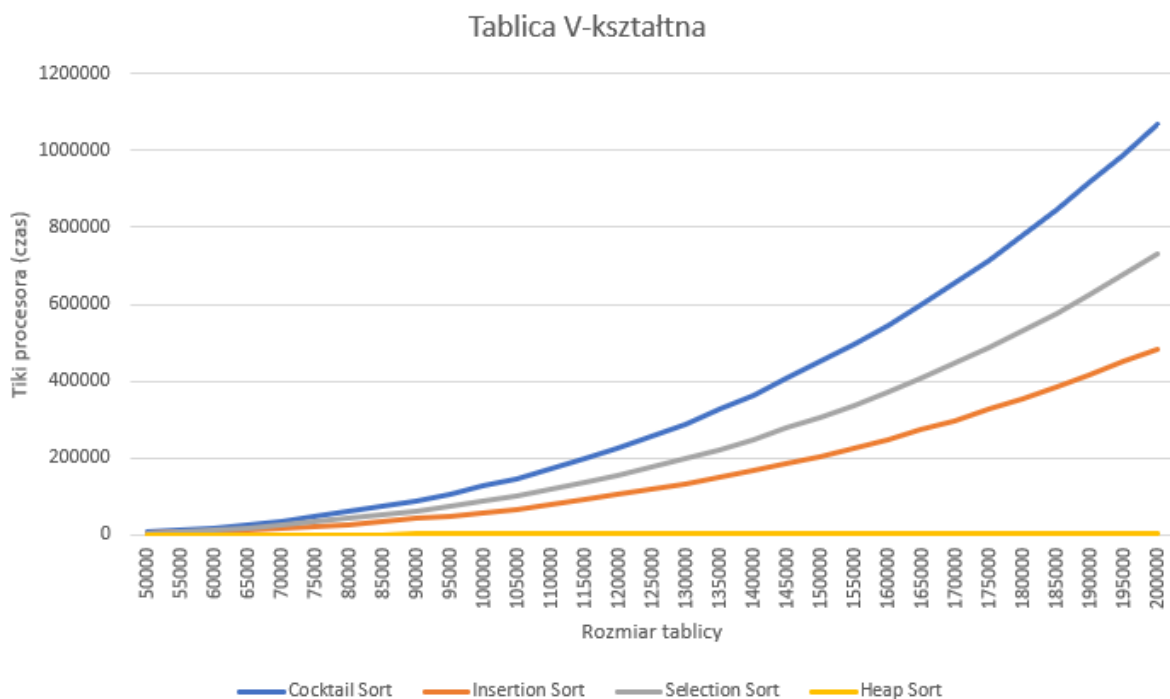


Ze względu na obraną skalę, oraz zbliżone wyniki, dane dotyczące Cocktail Sort, Insertion Sort, oraz Heap Sort zlewają się w jedną całość. W przypadku tablic stałych najskuteczniejsze okazał się Cocktail

Sort, a najgorszy dla Selection Sort, dokładnie tak samo jak w przypadku tablicy stałej. Heap Sort z powodu obranej skali wygląda na funkcję liniową, jednak w rzeczywistości nią nie jest.



W przypadku tablicy malejącej najskuteczniejszy okazał się Heap Sort, a najgorszym Coctail Sort. Heap Sort z powodu obranej skali wygląda na funkcję liniową, jednak w rzeczywistości nią nie jest.



Najgorzej w przypadku tablicy V-kształtnej najgorzej działa Coctail Sort, a najlepiej Heap Sort.

Heap Sort z powodu obranej skali wygląda na funkcję liniową, jednak w rzeczywistości nią nie jest.

Podsumowując wyniki eksperymentu możemy zauważyć, że w większości przypadków najlepiej poradził sobie Heap Sort (  $O(n \cdot \log(n))$  ), będąc najszybszym algorytmem w przypadku sortowania tablicy losowej, malejącej, oraz V-kształtnej. Zaraz po niej jest Sortowanie Koktajlowe (  $O(n^2)$  ), które najlepiej poradziło sobie z tablicą stałą, oraz rosnącą. Najwolniejszym algorytmem dla tablic rosnących i stałych okazał się Selection Sort, natomiast dla malejących, losowych i V-kształtnych był to Cocktail Sort.