

Projekt 3 - Sprawozdanie

Kacper Bok

Pz84473, W10 K31

Część I

W części I głównym celem było zbadanie oraz porównanie czasu działania stworzonych algorytmów 4 metod sortowania:

- *Insertion Sort, Selection Sort, Heap Sort, Cocktail Sort*

Dla tablicy liczb całkowitych (od 50k do 200k) generowanych w postaci:

- *Losowej, rosnącej, malejącej, stałej i V-kształtnej (V-shape)*

W tej części projektu łącznie przeprowadzono 20 eksperymentów, gdzie każdy z 4 algorytmów przyjął każdą z 5 rodzajów tablic. Na wykresach przedstawiono pomiary czasu sortowania danego algorytmu w zależności od rodzaju danych wejściowych. Po porównaniu danych z eksperymentu otrzymałem następujące wyniki:

- Użycie algorytmu Insertion Sort jest najbardziej korzystne dla tablicy typu Ascending (rosnącej) oraz stałej (Constant).

- Użycie algorytmu Selection Sort jest porównywalnie dobre dla wszystkich rodzajów tablic poza Constant Tab gdzie czas był znacznie niższy jedynie dla tego przypadku.

- Heap Sort – bez wątpienia najszybszy z wykorzystywanych algorytmów. Pomijając tablice typu Constant najlepiej sprawdził się dla Descending (malejącej) oraz Ascending Tab.

- Użycie Cocktail Sort jest bez wątpienia najlepsze dla tablicy typu Ascending oraz Constant.

Część II

W części II celem było zbadanie oraz porównanie czasu sortowania danego typu tablicy, w zależności od rodzaju algorytmu sortującego. Na wykresach przedstawiono pomiary czasu sortowania. Po porównaniu danych z eksperymentu otrzymałem następujące wyniki:

- Random Tab najlepiej sprawdza się dla algorytmu sortującego typu Heap Sort. Porównywalnie dla Insertion oraz Selection Sort. Najgorszy wynik został uzyskany używając algorytmu Random Sort.

- Descending Tab – sytuacja podobna jak przy użyciu Random Tab. Najlepszy czas uzyskał Heap Sort, porównywalnie dla Insertion oraz Selection Sort tym razem na korzyść Selection Sort.

Ascending Tab – wszystkie algorytmy sortujące poza Selection Sort sprawdzą się przy tego typu tablicy.

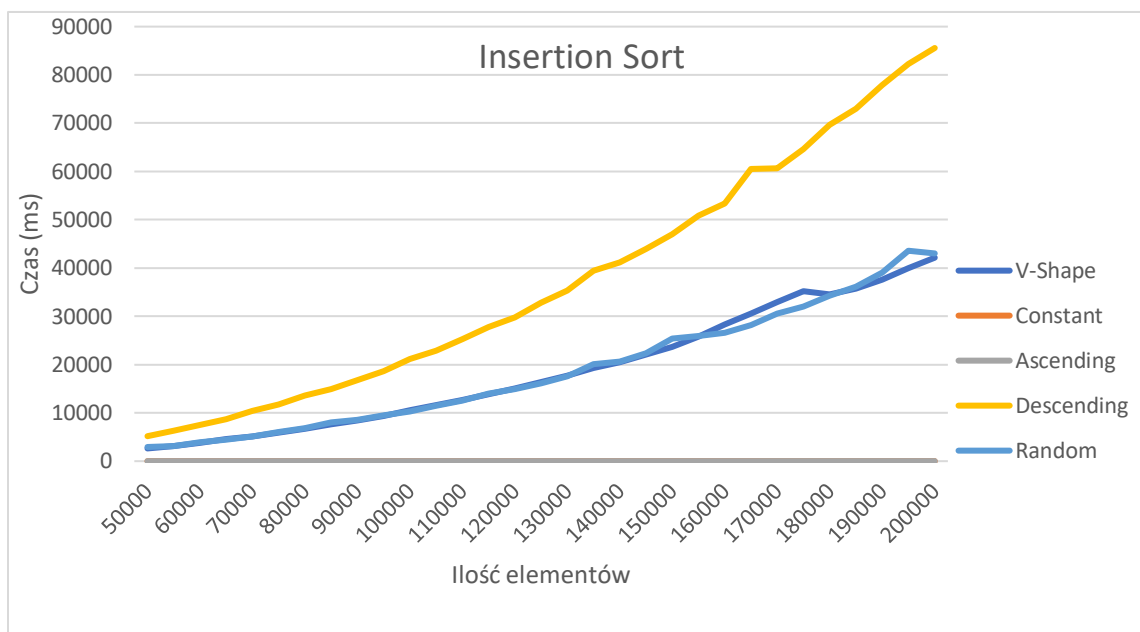
Constant Tab – wyniki mówią jednoznacznie - nie powinno się stosować przy sortowaniu typu Selection. Dla pozostałych trzech algorytmów sortujących wynik prawie, że identyczny.

Tablice typu V-Shape najszybciej posortuje algorytm typu Heap Sort. Następnie Insertion, Selection. Najgorszy wynik uzyskał algorytm sortujący Cocktail Sort.

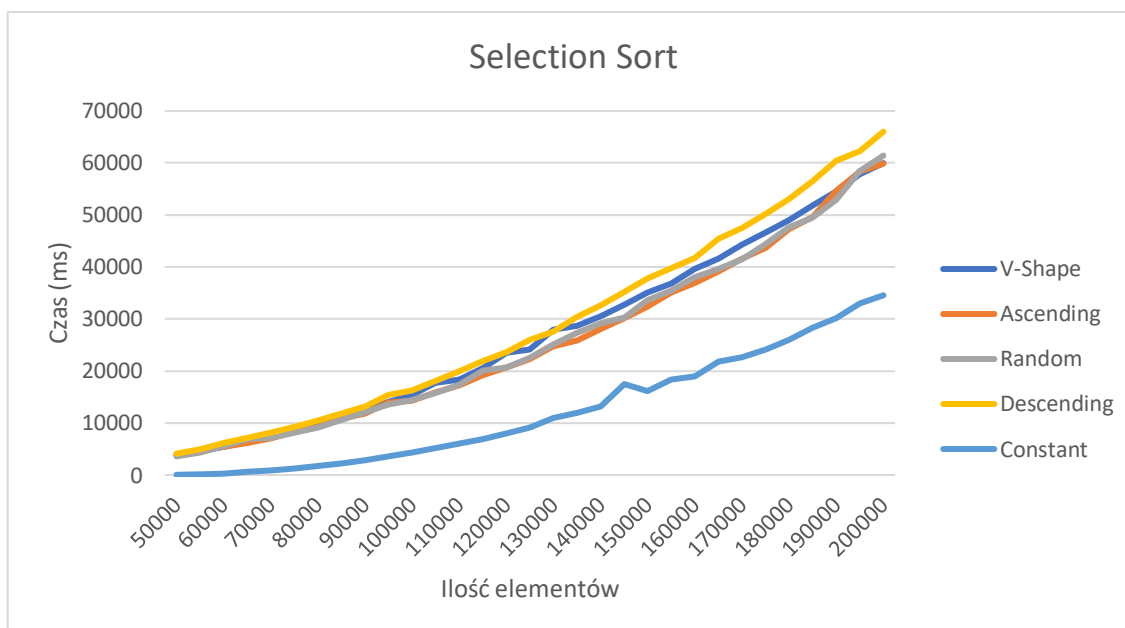
Uśredniając wyniki w największej ilości przypadków najszybszym algorytmem okazywał się algorytm typu **Heap Sort**.

Algorytm został napisany w języku C#, użyto programu Microsoft Visual Studio. Każdy algorytm wysyłał dane do pliku .csv zachowując go na dysku. W programie Excel zostały stworzone tabele oraz wykresy pokazujące czas/ilość operacji dla konkretnej liczby, które umożliwiają porównanie wyników dla każdego przypadku.

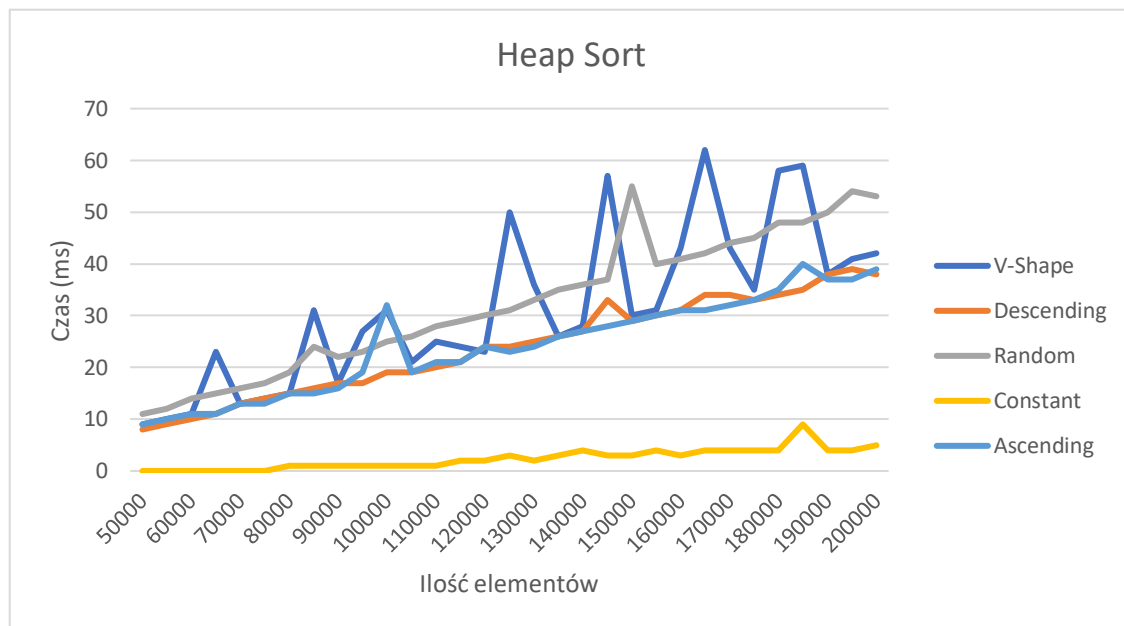




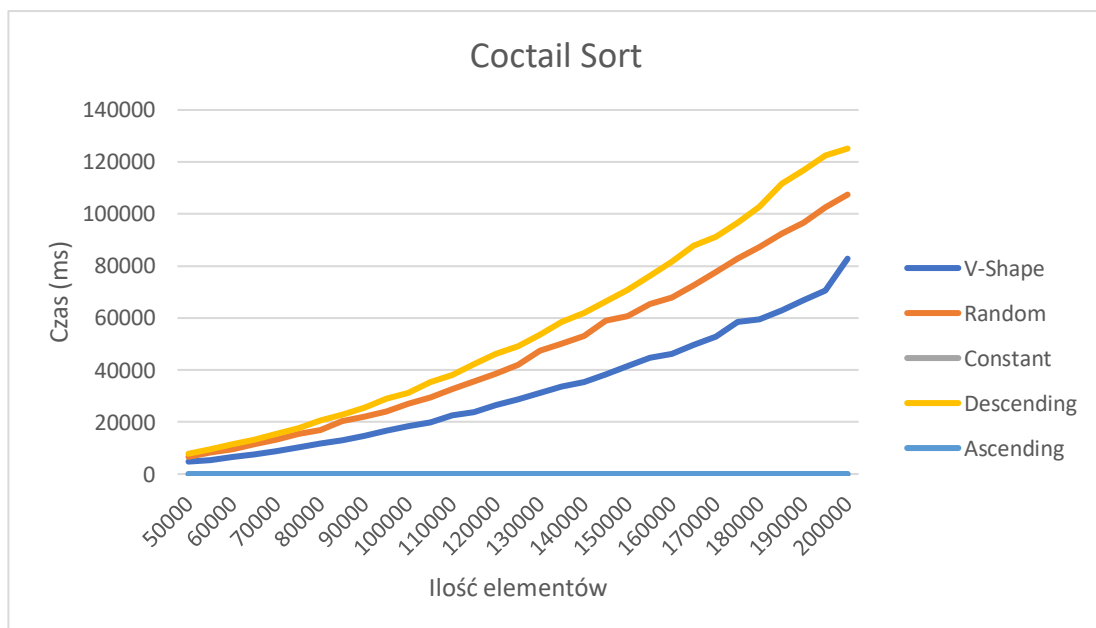
Insertion Sort okazał się najbardziej skuteczny dla Constant Tab oraz Ascending Tab. Biorąc pod uwagę pozostałe tablice pomiar jest porównywalny dla V-Shape Tab oraz Random Tab. Użycie Insertion Sort jest najmniej skuteczne dla tablicy malejącej (Descending Tab).



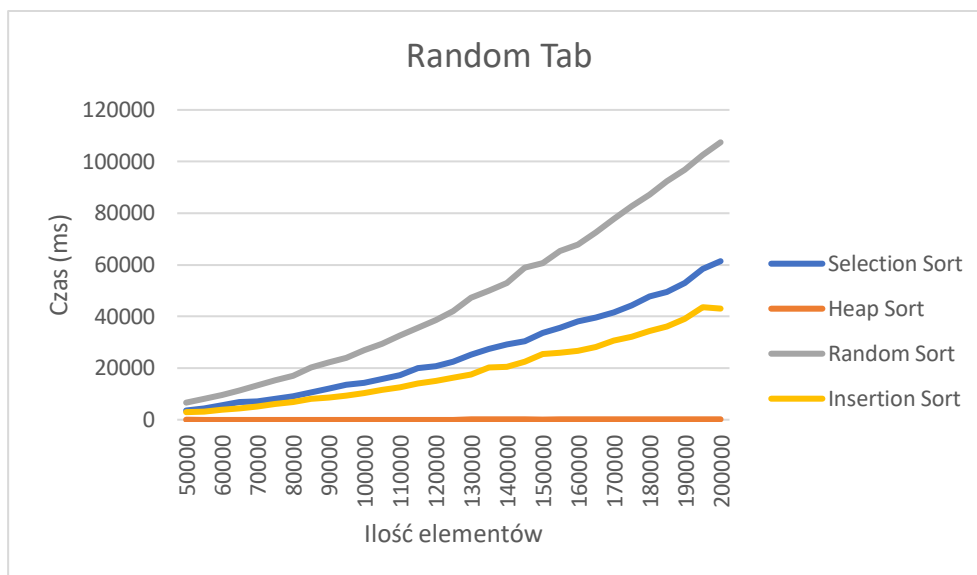
Użycie **Selection Sort** dla tablic typu V-Shape, Ascending, Descending, Random wypadło porównywalnie dobrze z minimalną przewagą dla tablicy Ascending Tab. Pomimo tego, że Constant Tab jest stały to czas sortowania takiej tablicy wzrastał przy użyciu Selection Sort.



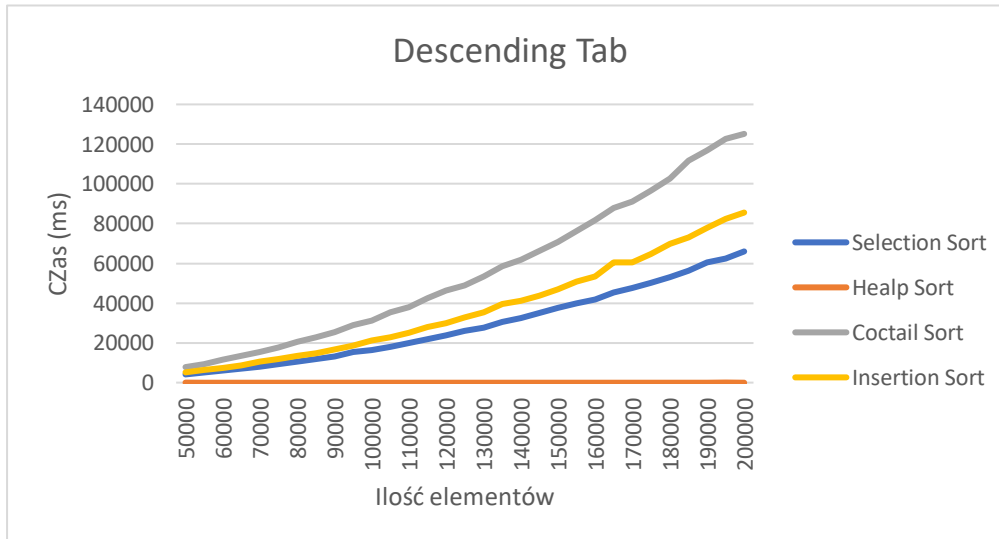
Heap Sort – najszybszy z wykorzystywanych algorytm sortowania tablic. Pomijając tablice typu Constant najlepiej sprawdził się dla V-Shape, Descending oraz Ascending Tab. Minimalnie gorzej wypadł dla tablicy typu Random.



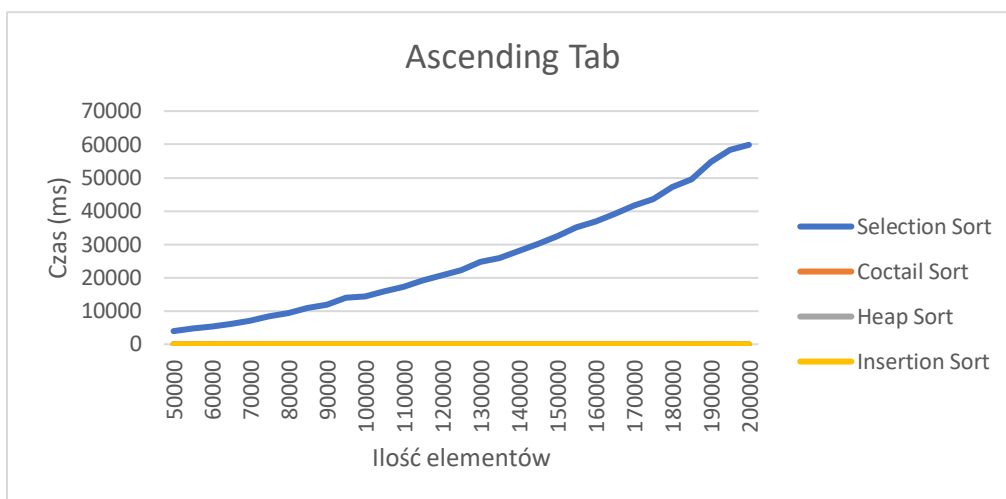
Coctail Sort – bez wątpienia najlepszy dla tablicy typu Ascending oraz Constant, najgorszy dla tablicy – Descending.



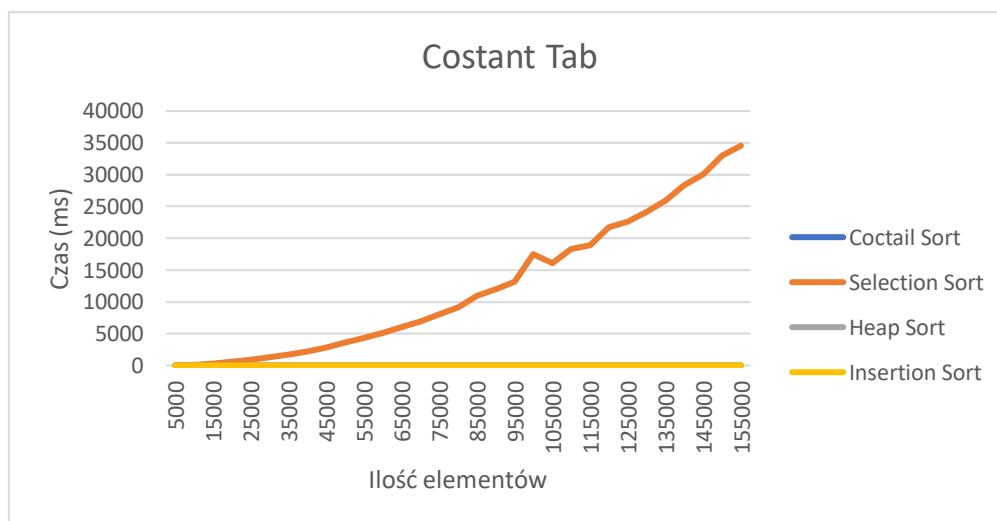
Random Tab najlepiej sprawdza się dla algorytmu sortującego typu Heap Sort. Porównywalnie dla Insertion oraz Selection Sort na korzyść Insertion Sort. Najgorszy wynik został uzyskany używając algorytmu Coctail Sort.



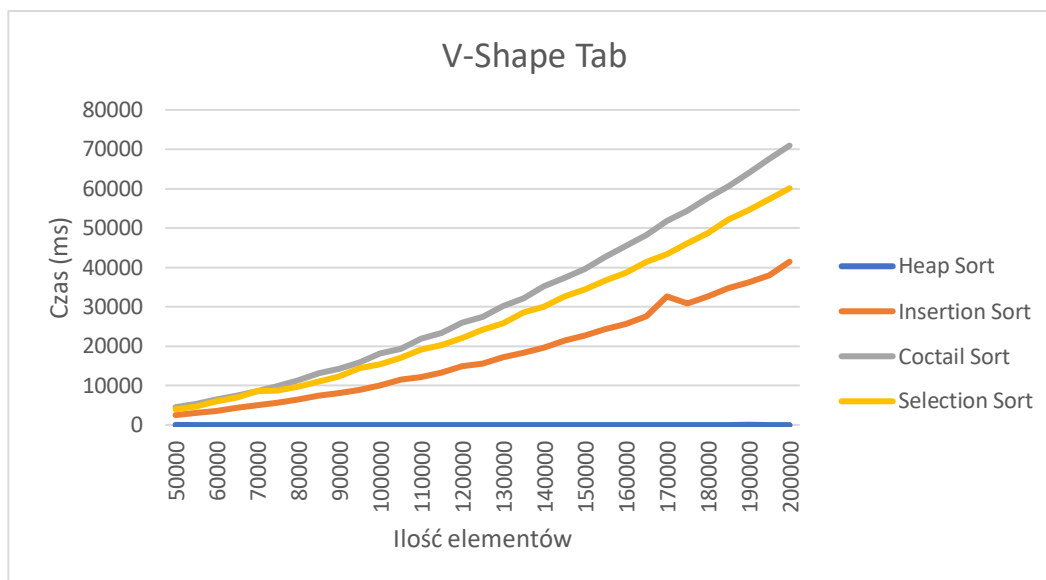
Descending Tab – sytuacja podobna jak przy użyciu Random Tab. Najlepszy czas uzyskał Heap Sort, porównywalnie dla Insertion oraz Selection Sort tym razem na korzyść Selection Sort.



Ascending Tab – wszystkie algorytmy sortujące poza Selection Sort sprawdzą się przy tego typu tablicy.



Constant Tab – wyniki mówią jednoznacznie - nie powinno się stosować przy sortowaniu typu Selection. Dla pozostałych trzech algorytmów sortujących wynik prawie, że identyczny.



Tablice typu **V-Shape** najszybciej posortuje algorytm typu Heap Sort. Następnie Insertion, Selection. Najgorszy wynik uzyskał algorytm sortujący Coctail Sort.