

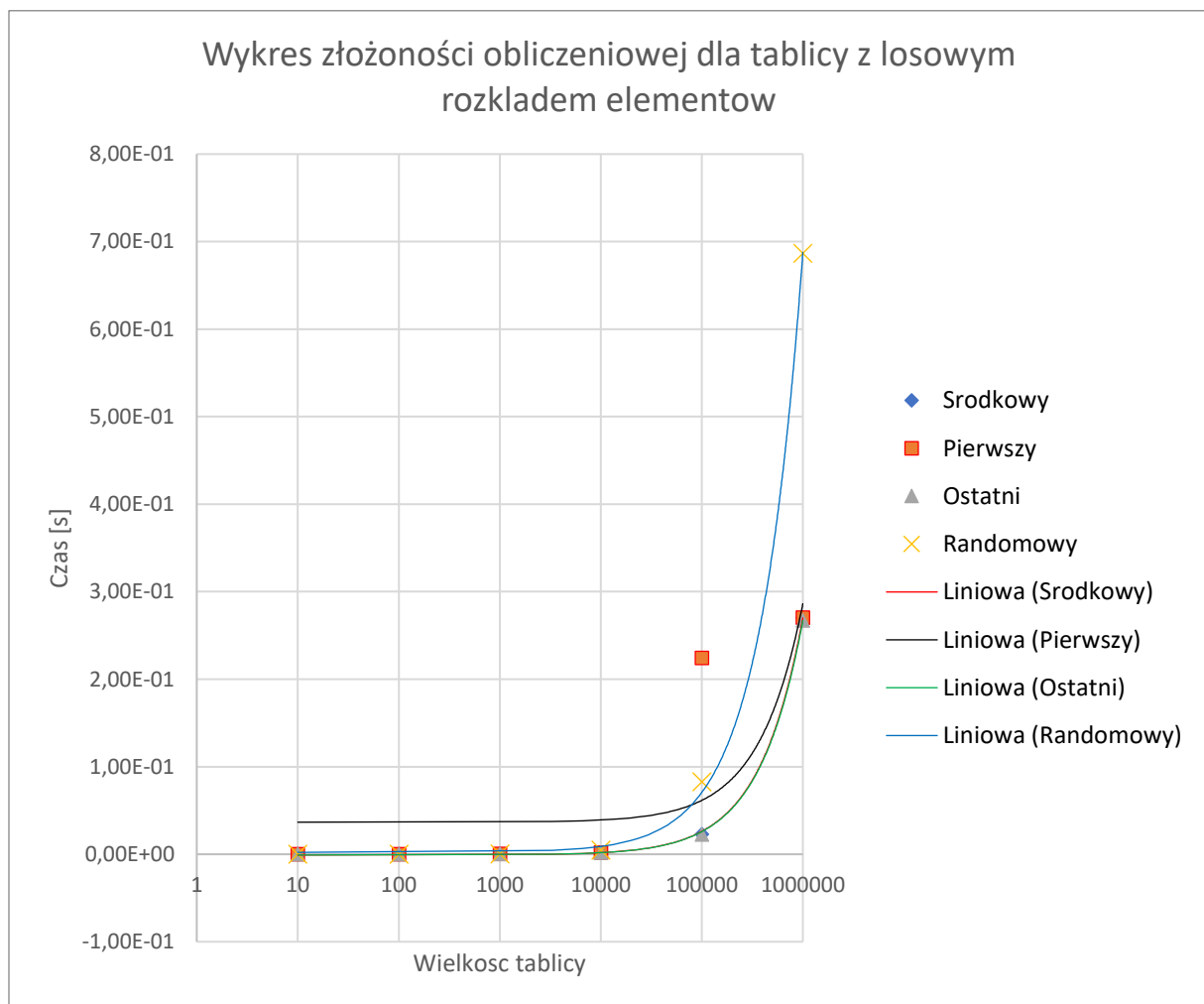
Sprawozdanie

Lab 5. Quicksort

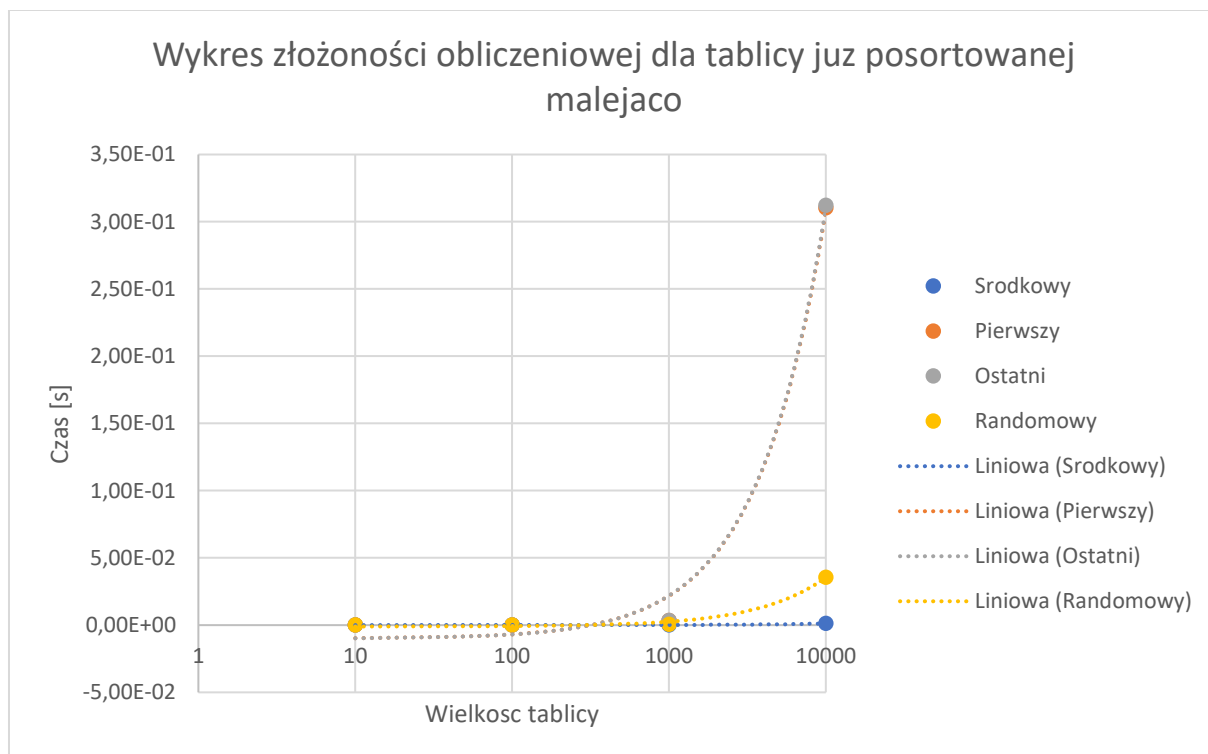
Implementacja sortowania szybkiego dla tablicy stworzonej do poprzednich zadań w odpowiednich plikach. Poszczególne czasy dla 3 wariantów wybrania piwota oraz 6 wariantów wielkości tablicy która ulega sortowaniu, również 3 przypadki rozrzucenia elementów w tablicy. Złożoność obliczeniowa sortowania szybkiego wynosi $O(n \log n)$, w pesymistycznym przypadku, czyli wtedy gdy tablice podczas sortowania partycjonujemy na 2 znacząco różniące się od siebie wielkościami części, złożoność oczekiwana wynosi $O(n^2)$. W przypadku dużych rozmiarów tablic ilość wywołań rekursywnych algorytmu może doprowadzić do przepełnienia stosu wywołań a tym samym do zatrzymania pracy programu, tak też się stało w mojej implementacji.

Na każdym wykresie pokazano 3 możliwości wyboru piwota: środkowy, pierwszy oraz ostatni.

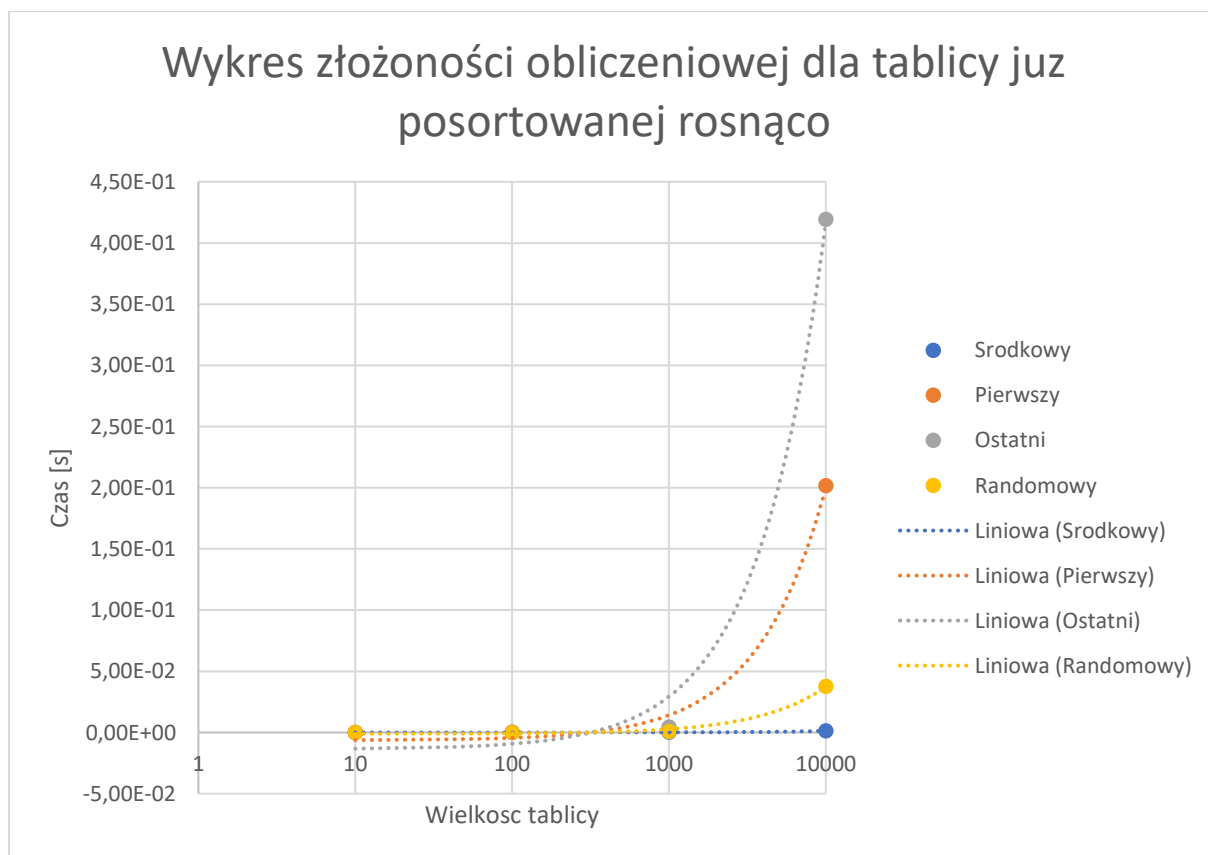
Poszczególne wyniki przedstawiono na wykresach poniżej:



Rysunek 1 Wykres czasu wykonywania algorytmu sortowania szybkiego w zależności od ilości elementów do sortowania. Dla piwota w środku i na koncu wykresy pokrywają się.



Rysunek 2 Wykres złożoności obliczeniowej dla tablicy już posortowanej malejąco. Wykresy dla pivotu wybranego na skrajach rosną o wiele szybciej dlatego wykres dla pivotu wybranego w środku wygląda w tym przedziale na stały.



Rysunek 3 Wykres złożoności obliczeniowej dla tablicy już posortowanej rosnąco.

Wnioski

Na wykresie pierwszym najlepiej widać, że złożoność obliczeniowa kształtuje się w wykres logarytmiczny, toteż można stwierdzić, że dla przypadku średniego złożoność obliczeniowa sortowania szybkiego to $O(n \log n)$.

Biorąc skrajne elementy jako miejsca partycjonowania tablicy do sortowania złożoność zwiększa się coraz bardziej przypominając wykres funkcji kwadratowej. Na Ostatnim wykresie nie jest to ukazane ale ustawiając pivot jako środkowy element tablicy wykres faktycznie staje się logarytmiczny a dla pozostałych przypadków staje się kwadratowy, jednak przez przepełnienie stosu wywołań funkcji bardzo trudne stawało się zebranie wyników dla większej ilości elementów do sortowania.

Zgodnie z oczekiwaniami algorytm dla dobrania odpowiedniego pivotu może stać się optymalny, lecz wraz ze złym wyborem zwiększa się złożoność obliczeniowa.

Dla przypadku w którym tablica jest już posortowana, widać najlepiej różnice w złożoności obliczeniowej dla wyboru każdego pivotu, najlepiej sprawdza się wtedy wybranie elementu dokładnie w środku tablicy, a najgorzej-wyбір ostatniego elementu. Zgodnie z oczekiwaniami w najgorszym przypadku wykres staje się zbliżony do funkcji kwadratowej, a w średnim – do wykresu logarytmicznego.