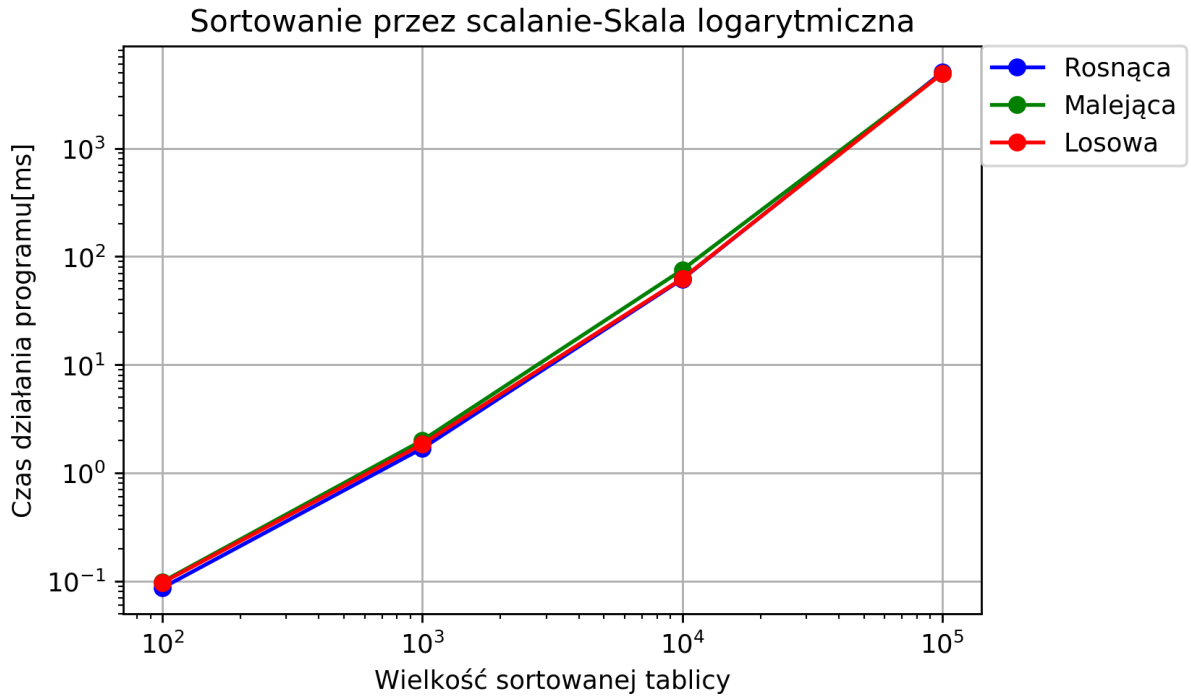
Sprawozdanie

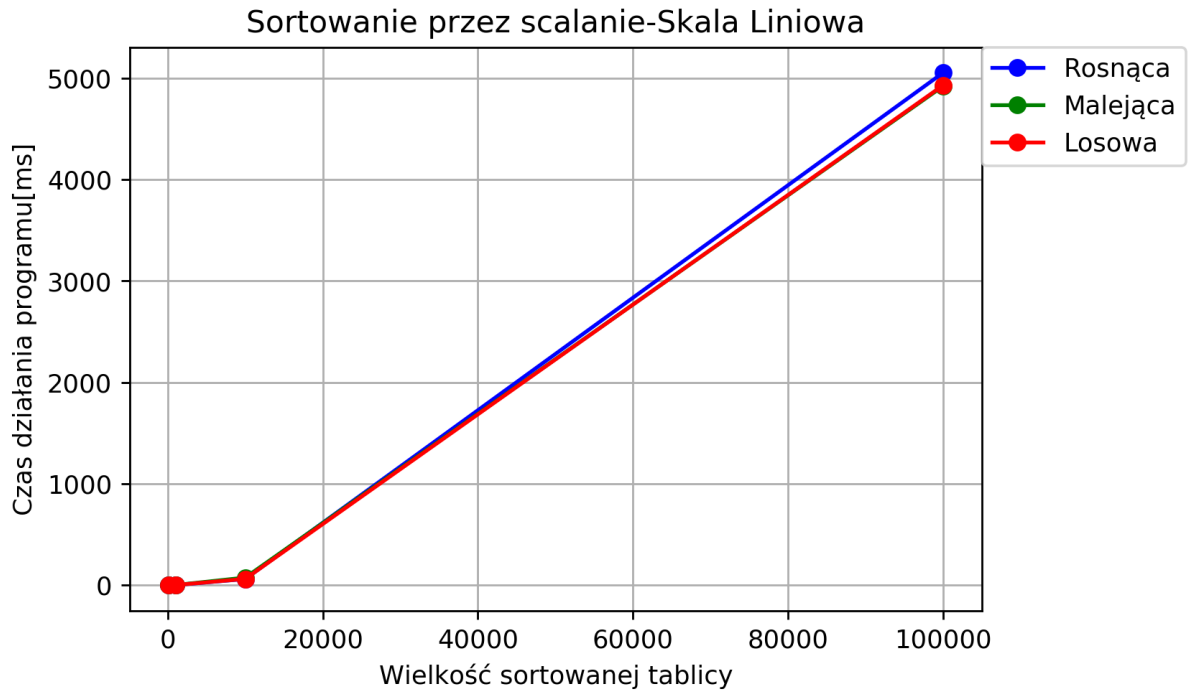
Lab6(Mergesort)

Zadanie polegało na zaimplementowaniu sortowania przez scalanie a następnie zmierzenie czasu działania algorytmu w zaleznosci od ilości elementów w tablicy oraz stopnia posortowania jeszcze nie sortowanej tablicy. Utrudnienie polegało jednak na implementacji algorytmu w programie z poprzednich zajęć stworzonym przez kolegę z grupy. Program Arkadiusza Zemy z implementacją sortowania przez scalanie w osobnych plikach, tutaj wykresy przedstawiające przybliżoną złożoność obliczeniową algorytmu.



Rysunek 1Wykres złożoności obliczeniowej mergesort w skali logarytmicznej dla tablicy posortowanej rosnąco, malejąco oraz z losowym ustawieniem elementów.

Na wykresie w skali logarytmicznej widać jak wykres kreuje się dla większych tablic, coraz bardziej zaczyna przypominać funkcję liniową. Zachowanie takie jest jak najbardziej zrozumiałe i oczekiwane.



Rysunek 2Rysunek 1Wykres złożoności obliczeniowej mergesort w skali liniowej dla tablicy posortowanej rosnąco, malejąco oraz z losowym ustawieniem elementów.

Wykres w skali liniowej dobrze pokazuje początek wykresu czyli złożoność obliczeniową dla małych tablic, tutaj jest z kolei widoczne podobieństwo do wykresu funkcji nlog(n).

Wnioski

- Spodziewana złożoność obliczeniowa dla algorytmu sortowania przez scalanie wynosi O(n)=n*log(n), a przyglądając się wykresom powyżej można stwierdzić, że taki przebieg przypominają. Stąd wniosek, że algorytm działa ze skutkiem oczekiwanym.
- Sortowanie przez scalanie w przypadku pesymistycznym czyli dla posortowanych już rosnąco elementów działa najwolniej jednak jedynie dla bardzo dużych zbiorów.
- Dla małych i nieuporządkowanych w żaden sposób tablic algorytm działa z kolei najszybciej.