

Rzeszów, 14.12.2021

SPRAWOZDANIE

Jakub Misiło

Inżynieria i Analiza Danych

I rok, grupa V

1. Podstawy teoretyczne

a) **Sortowanie przez wstawianie** - jeden z najprostszych algorytmów sortowania, którego zasada działania odzwierciedla sposób w jaki ludzie ustawiają karty – kolejne elementy wejściowe są ustawiane na odpowiednie miejsca docelowe. Jest efektywny dla niewielkiej liczby elementów, jego złożoność wynosi $O(n^2)$. Pomimo tego, że jest znacznie mniej wydajny od algorytmów takich jak np. quicksort, posiada pewne zalety:

- liczba wykonanych porównań jest zależna od liczby inwersji w permutacji, dlatego algorytm jest wydajny dla danych wstępnie posortowanych,
- jest wydajny dla zbiorów o niewielkiej liczebności,
- jest stabilny.

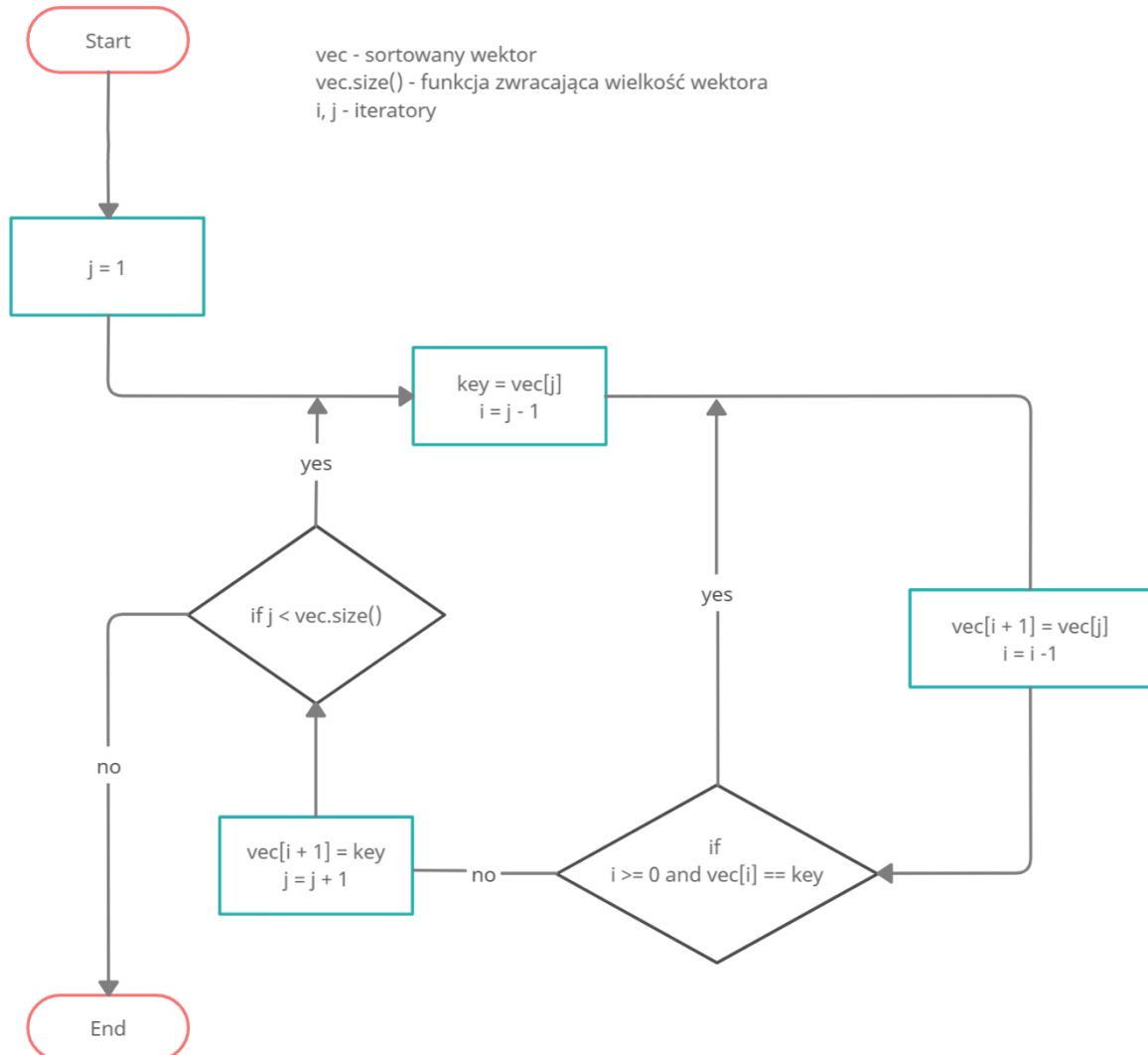
b) **Quicksort** - jest uważany za najszybszy algorytm dla danych losowych. Zasada jego działania opiera się o metodę **dziel i zwyciężaj**. Zbiór danych zostaje podzielony na dwa podzbiory i każdy z nich jest sortowany niezależnie od drugiego.

Dla zadanej tablicy $a[l..p]$ wybieramy element $v=a[l]$ i przeszukujemy resztę tablicy (tzn. $a[l+1..p]$) tak długo, aż nie znajdziemy elementu większego niż $a[l]$. Następnie przeszukujemy tą tablicę od strony prawej póki nie znajdziemy elementu nie większego niż $a[l]$. Gdy to osiągniemy, zamieniamy miejscami te dwa elementy i zaczynamy cały proces od początku. Algorytm działa tak długo, aż wskaźnik poruszający się w lewo i wskaźnik poruszający się w prawo spotkają się. Należy wówczas zamienić element $v=a[l]$ z ostatnim elementem lewej części tablicy.

Mimo, że w najgorszym przypadku algorytm ma złożoność kwadratową, jest on bardzo często stosowany. Powodem tego jest niska- liniowologarytmiczna, złożoność w średnim przypadku.

2. Schematy blokowe

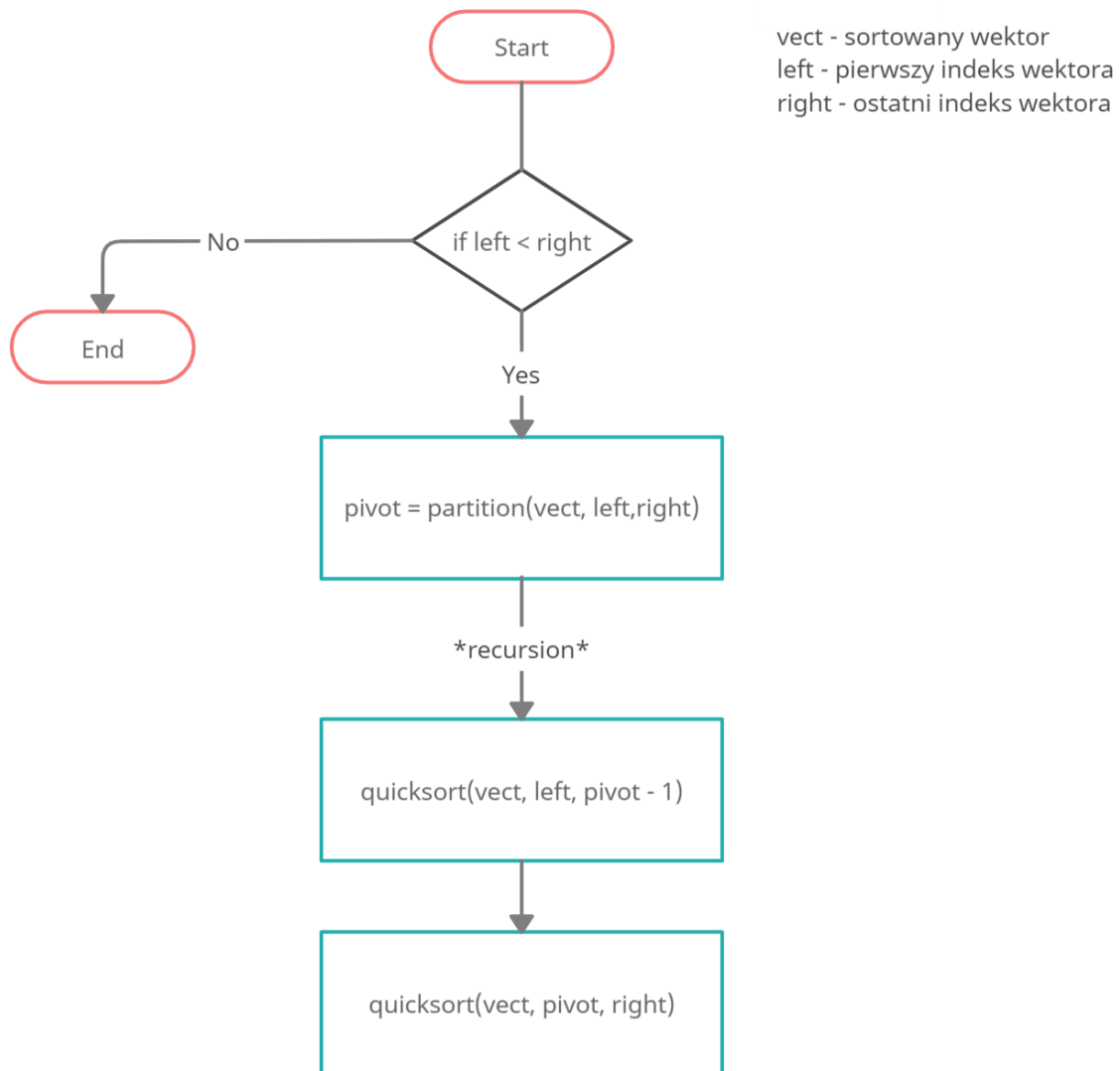
a) Sortowanie przez wstawianie



b) Quicksort

- Funkcja quicksort

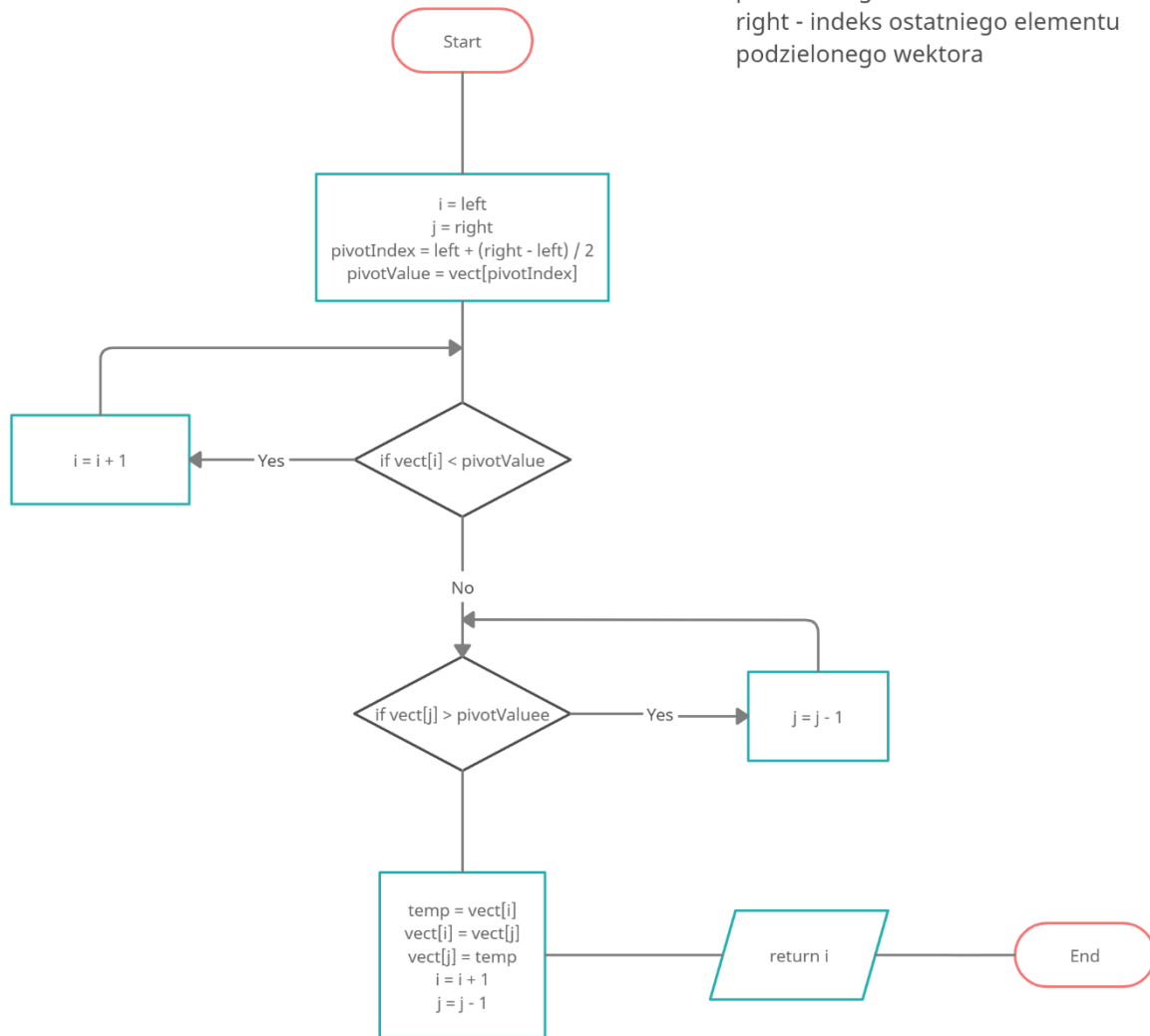
Quicksort (vect, left, right)



- Funkcja partition

partition(vect, left, right)

vect - sortowany wektor
 left - indeks pierwszego elementu z
 podzielonego wektora
 right - indeks ostatniego elementu
 podzielonego wektora



3. Pseudokod:

a) Sortowanie przez wstawianie:

```
j = 1
while j < vec.size():
    key = vec[j]
    i = j - 1

    while i >= 0 and vec[i] == key:
        vec[i + 1] = vec[j]
        i = i - 1
    vec[i + 1] = key
    j = j + 1
```

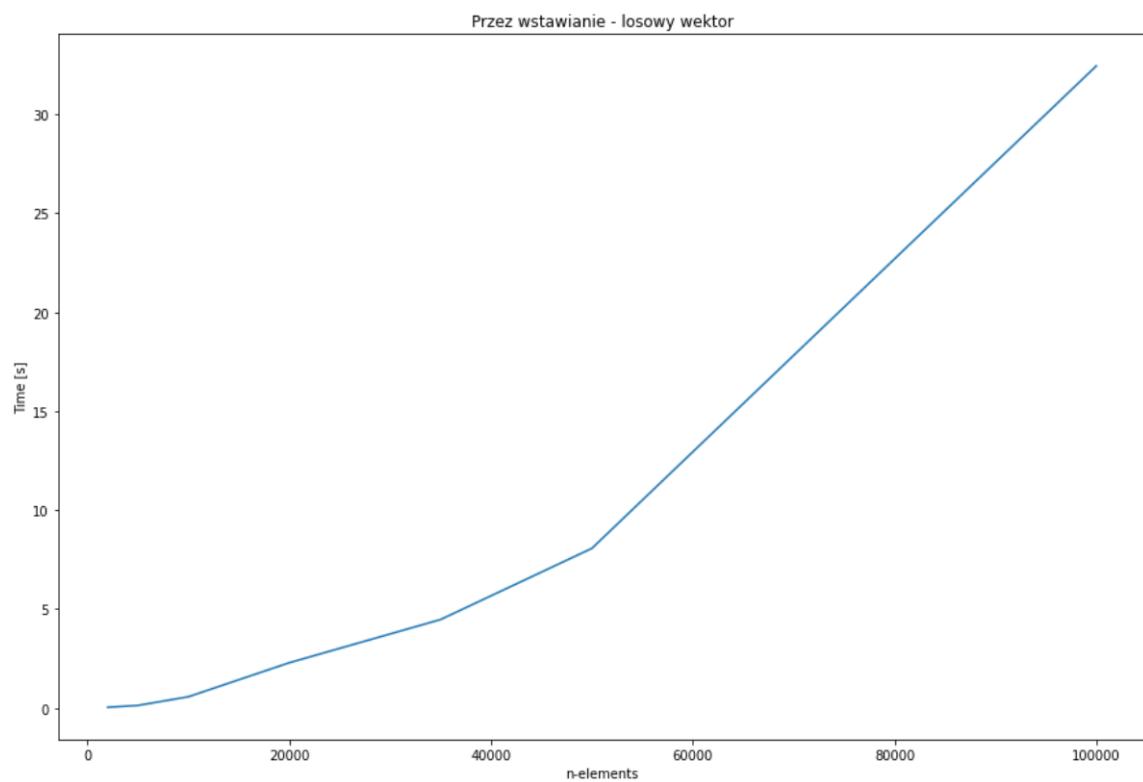
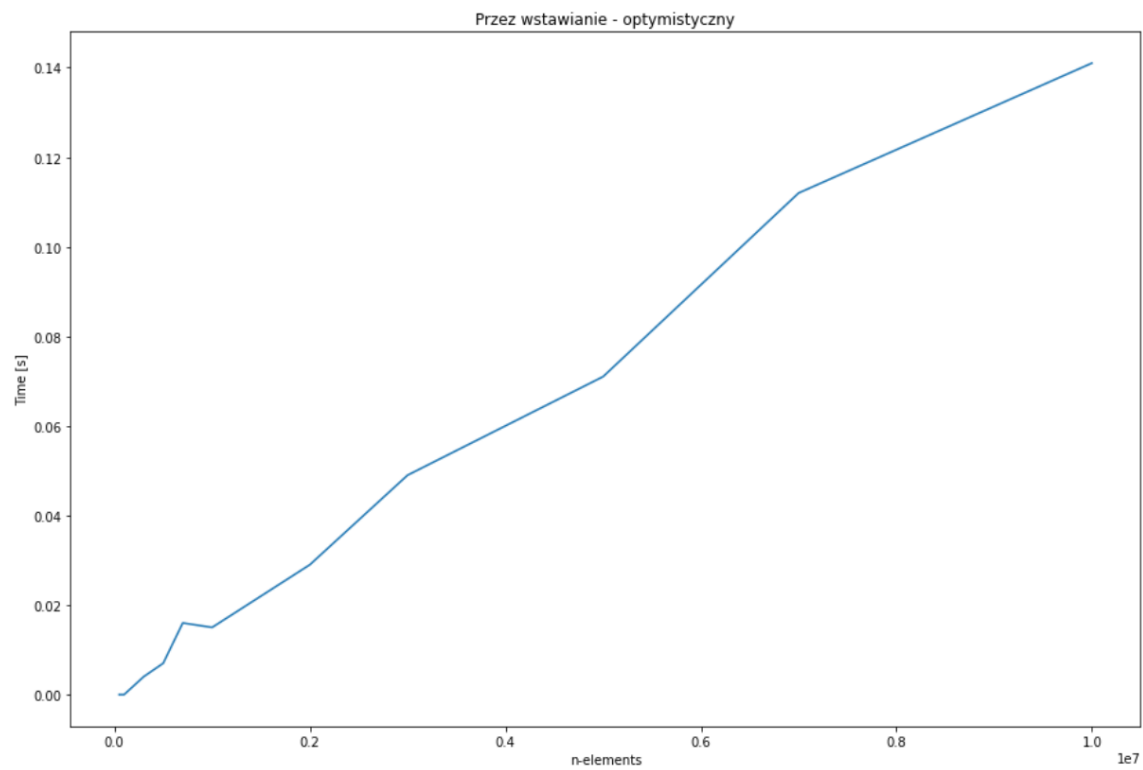
b) Quicksort

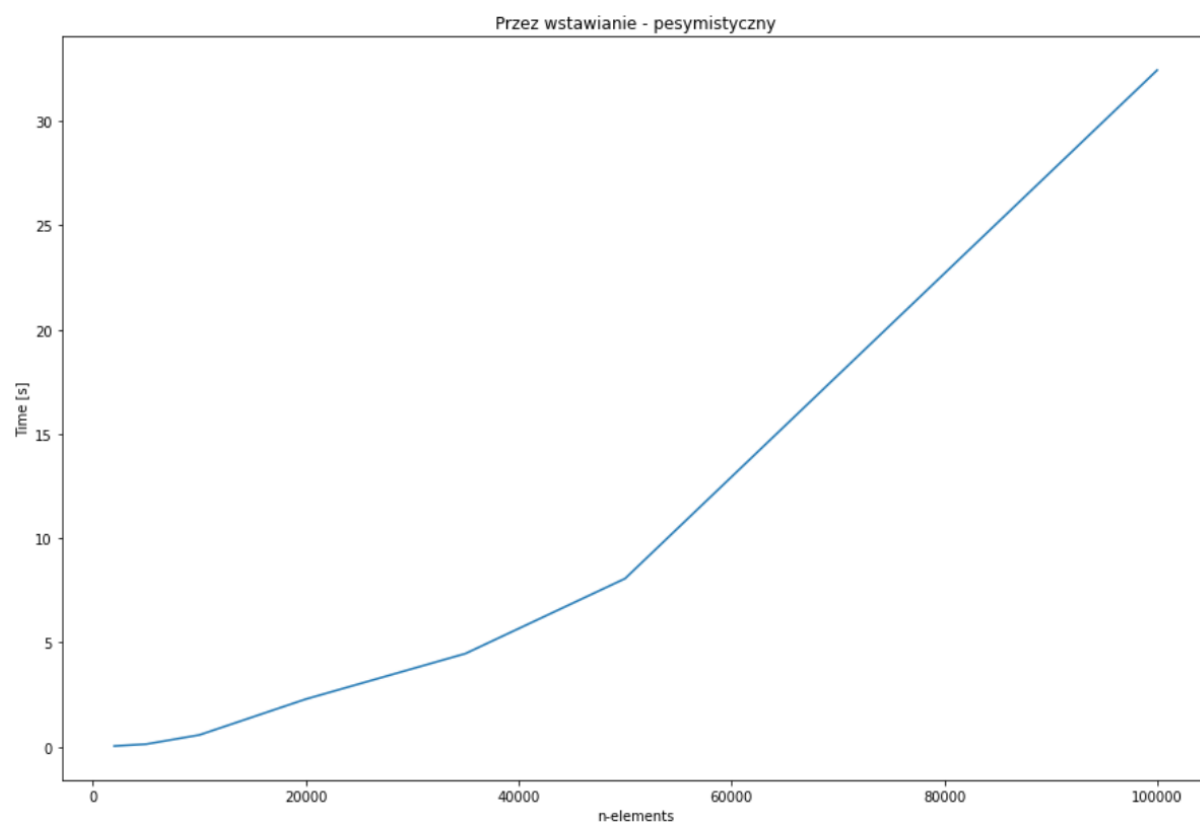
```
function quicksort(vect, left, right):
    if left < right:
        pivot = partition(vect, left, right)
        quicksort(vect, left, pivot - 1)
        quicksort(vect, pivot, right)

function partition(vect, left, right):
    i = left
    j = right
    pivotIndex = left + (right - left) / 2
    pivotValue = vect[pivotIndex]
    while vect[i] < pivotValue:
        i = i + 1
    while vect[j] > pivotValue:
        j = j - 1
    temp = vect[i]
    vect[i] = vect[j]
    vect[j] = temp
    i = i + 1
    j = j - 1
    return i
```

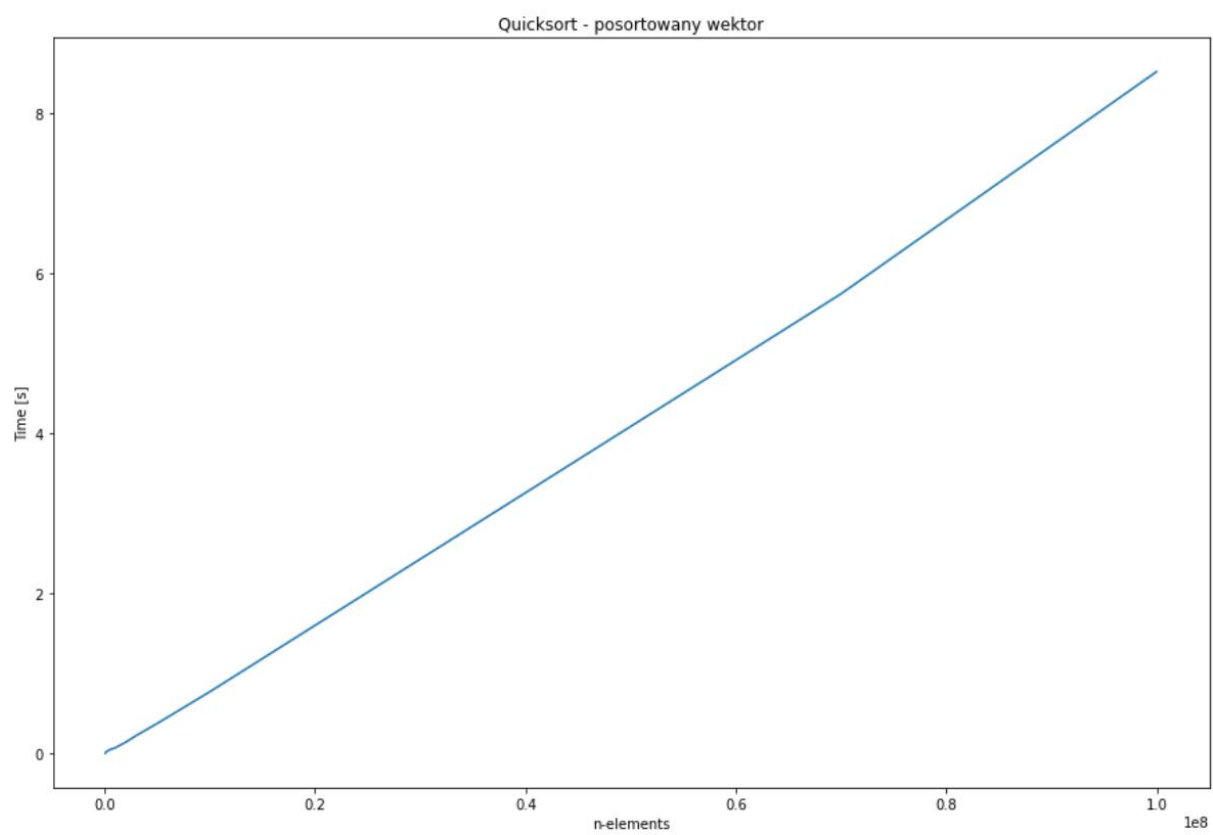
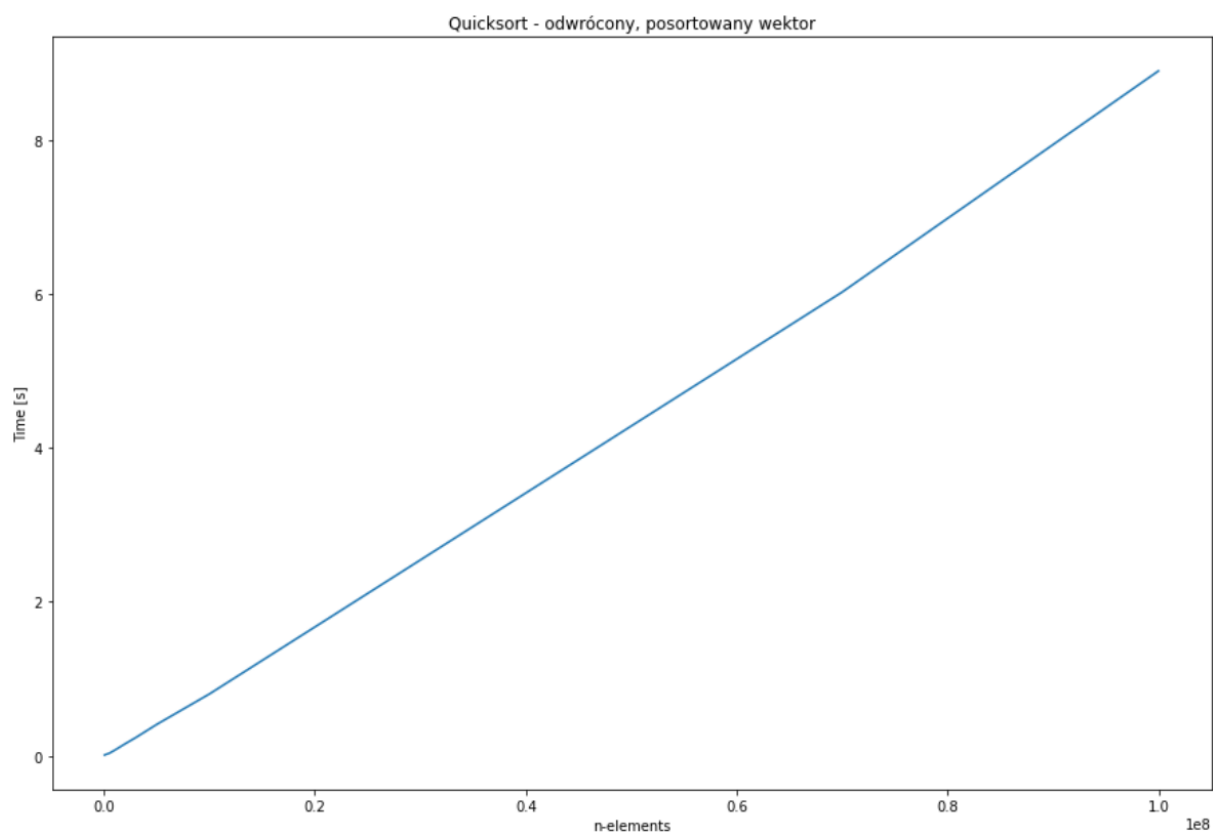
4. Złożoność czasowa:

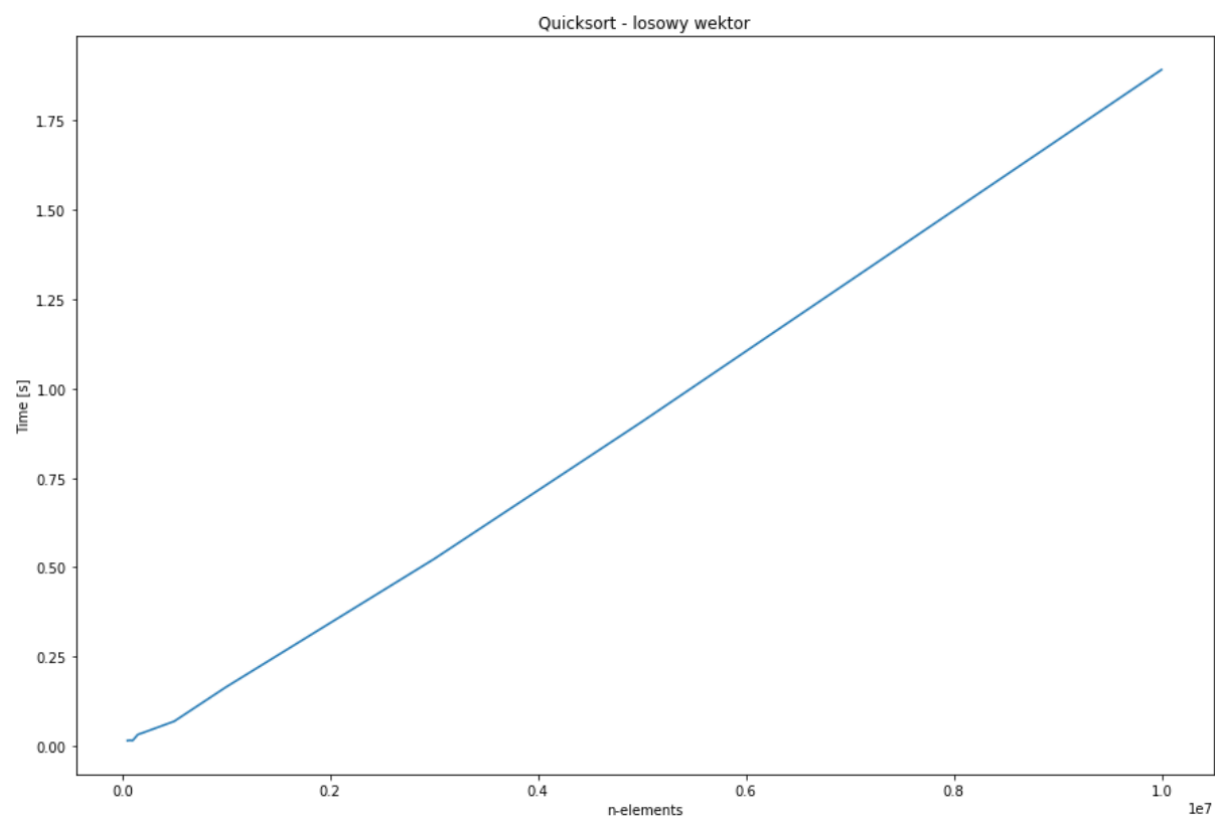
a) Sortowanie przez wstawianie





b) Quicksort





[Link do repozytorium*](#)