

Rzeszów, 30.10.2021

# **SPRAWOZDANIE**

Jakub Misiło

Inżynieria i Analiza Danych

I rok, grupa V

## Opis problemu:

Dla ciągu zawierającego wartości całkowite, znajdź malejący podciąg o największej długości.

## Algorytm:

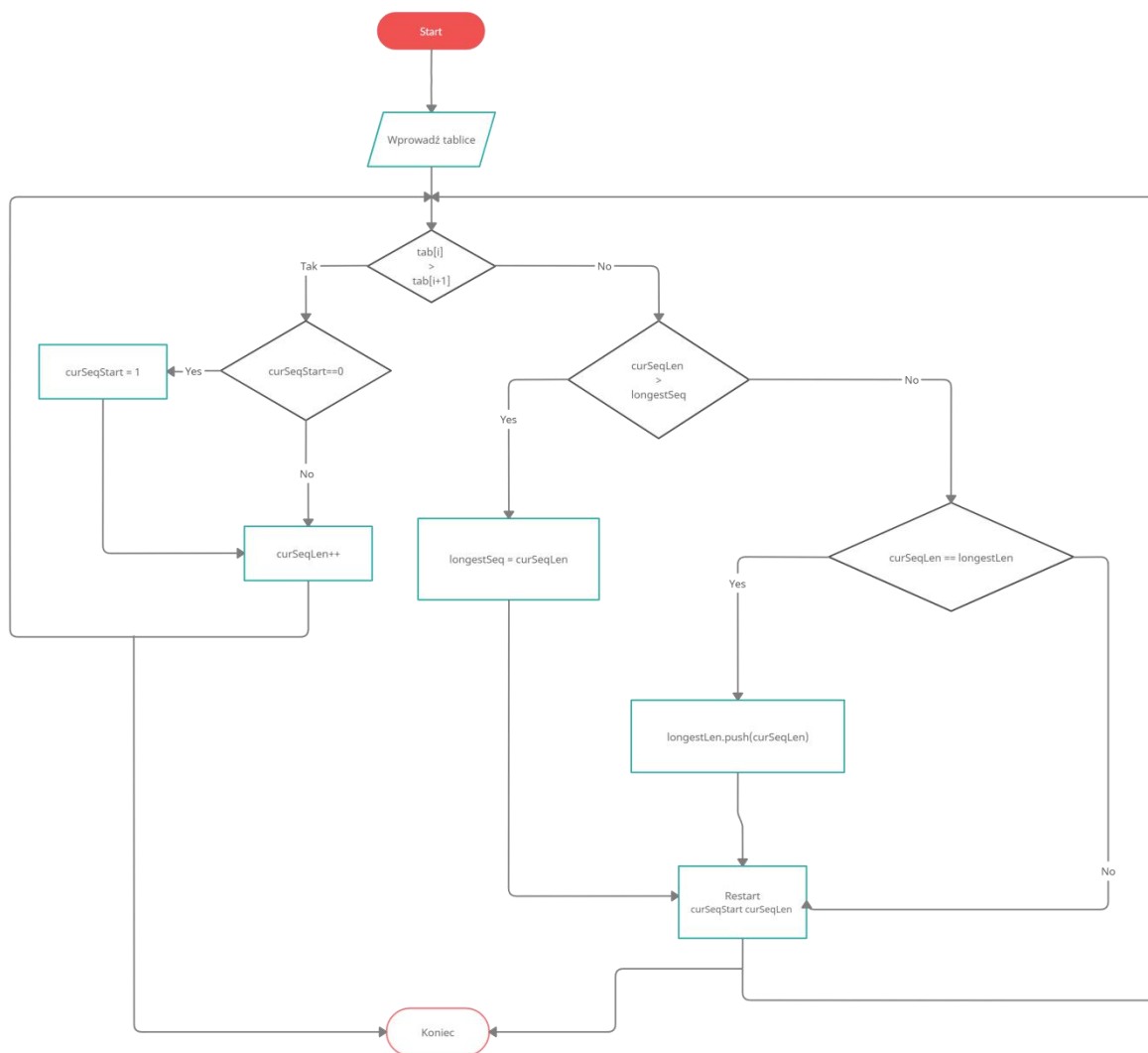
Rozpoczynamy od wprowadzenia wektora – tablicy jednowymiarowej. Uruchamiamy pętlę for o zakresie długości wprowadzonego wektora. Jeśli element wektora o indeksie  $i$  jest większy od elementu o indeksie  $i + 1$  to zwiększamy długość aktualnego ciągu o 1. W przeciwnym wypadku porównujemy długość ostatniego ciągu z póki co najdłuższym znalezionym ciągiem malejącym. Jeśli aktualny jest najdłuższym ciągiem - przypisujemy go do zmiennej. Jeśli jego długość jest równa długości najdłuższego ciągu - indeks początkowy aktualnego ciągu zostaje dodany do wektora z indeksem początkowym najdłuższego ciągu. Po zakończeniu wykonywania pętli zostają przedstawione rezultaty.

## Pseudokod:

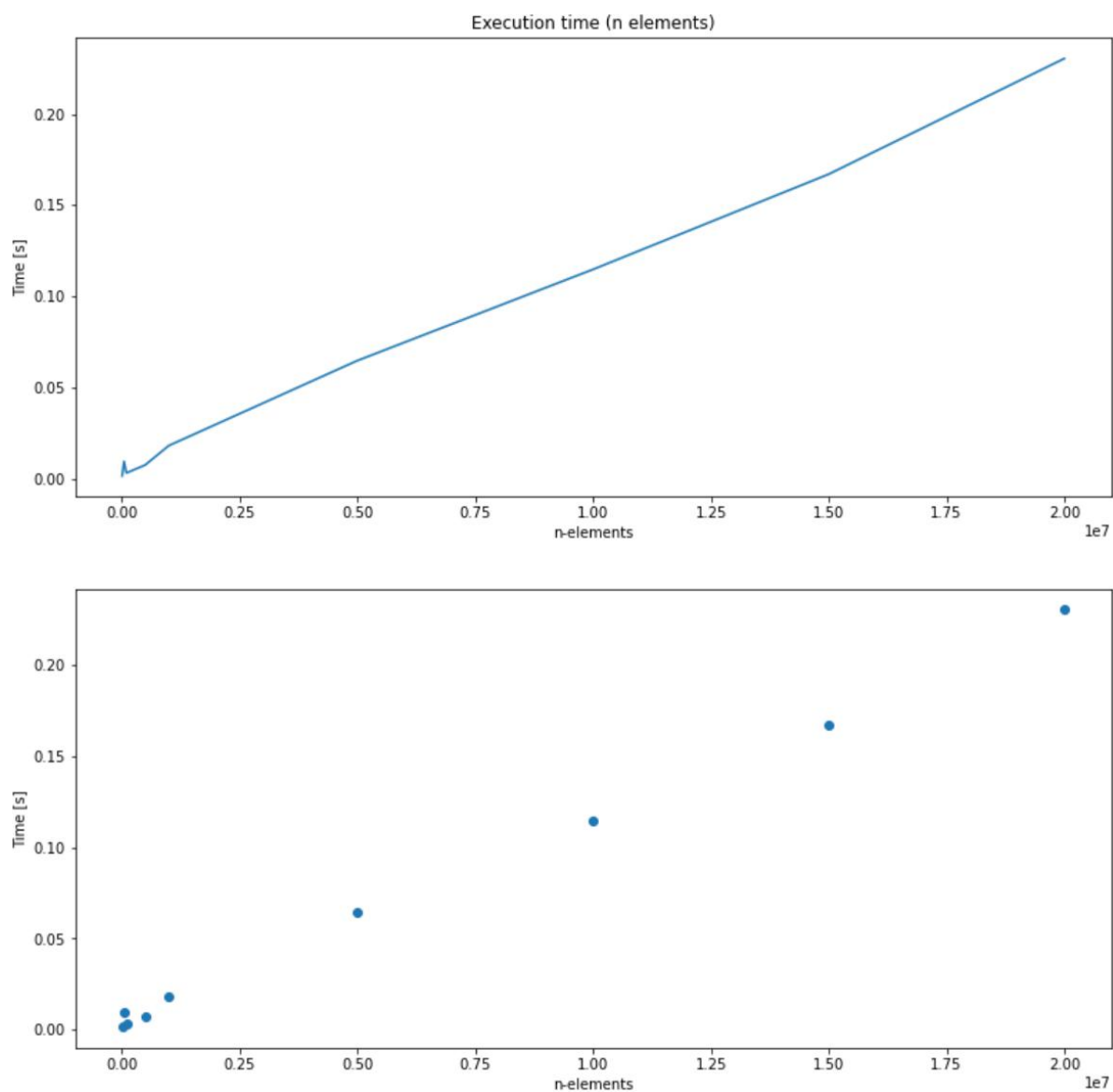
```
vect = generateVector()
curSeqStart, curSeqLen, longestLen = 0
longestStart = []

for i < vect.size:
    if vect[i] > vect[i + 1]:
        if curSeqStart == 0:
            curSeqStart = i
        curSeqLen++
    else:
        if curSeqLen > longestLen:
            longestStart.clear()
            longestStart.push(curSeqStart)
            longestLen = curSeqLen
        else if curSeqLen == longestLen:
            longestStart.push_back(curSeqStart)
        curSeqStart = 0
        curSeqLen = 0
```

## Schemat blokowy:



Wykres złożoności czasowej:



Kod programu \*

Wyniki przeprowadzonych testów \*

