

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:

```
// curSeqLen - długość aktualnie porównywanego ciągu
// curSeqStart - indeks początkowy porównywanego ciągu
// longestLen - długość dotychczas najdłuższych ciągów
// longestStart - wektor z indeksami początkowymi dotychczas najdłuższych ciągów
// arr.push(a) - dodanie elementu a na ostatni indeks tablicy arr
// arr.clear() - wyczyszczenie tablicy arr
// generateVector() - funkcja generująca wektory
```

```
vect = generateVector()
```

```
curSeqStart, curSeqLen, longestLen = 0
```

```
longestStart = []
```

```
for i < vect.size:
```

```
    if vect[i] > vect[i + 1]:
```

```
        if curSeqStart == 0:
```

```
            curSeqStart = i
```

```
            curSeqLen == curSeqLen + 1
```

```
    else:
```

```
        if curSeqLen > longestLen:
```

```
            longestStart.clear()
```

```
            longestStart.push(curSeqStart)
```

```
            longestLen = curSeqLen
```

```
        else if curSeqLen == longestLen:
```

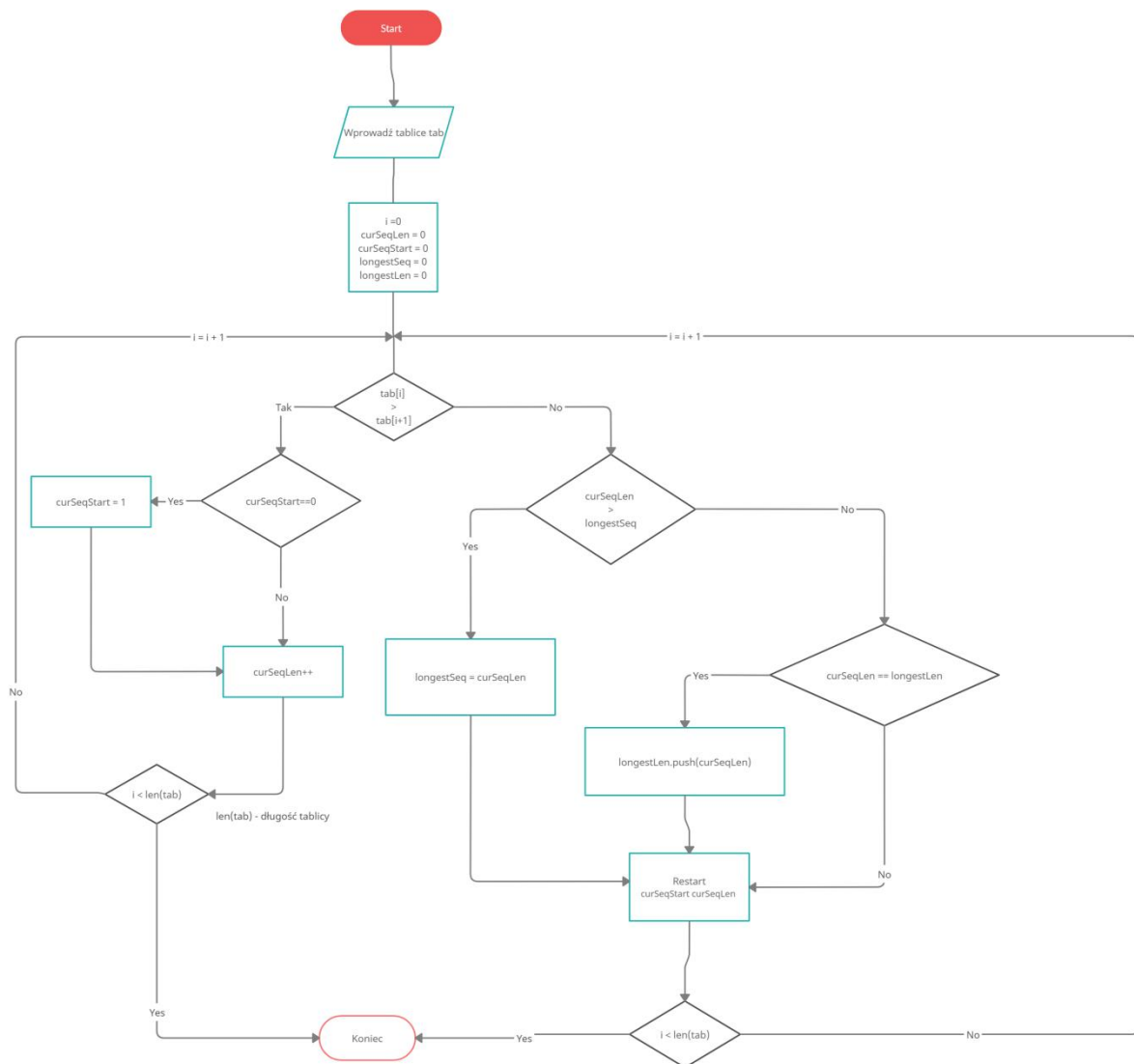
```
            longestStart.push(curSeqStart)
```

```
            curSeqStart = 0
```

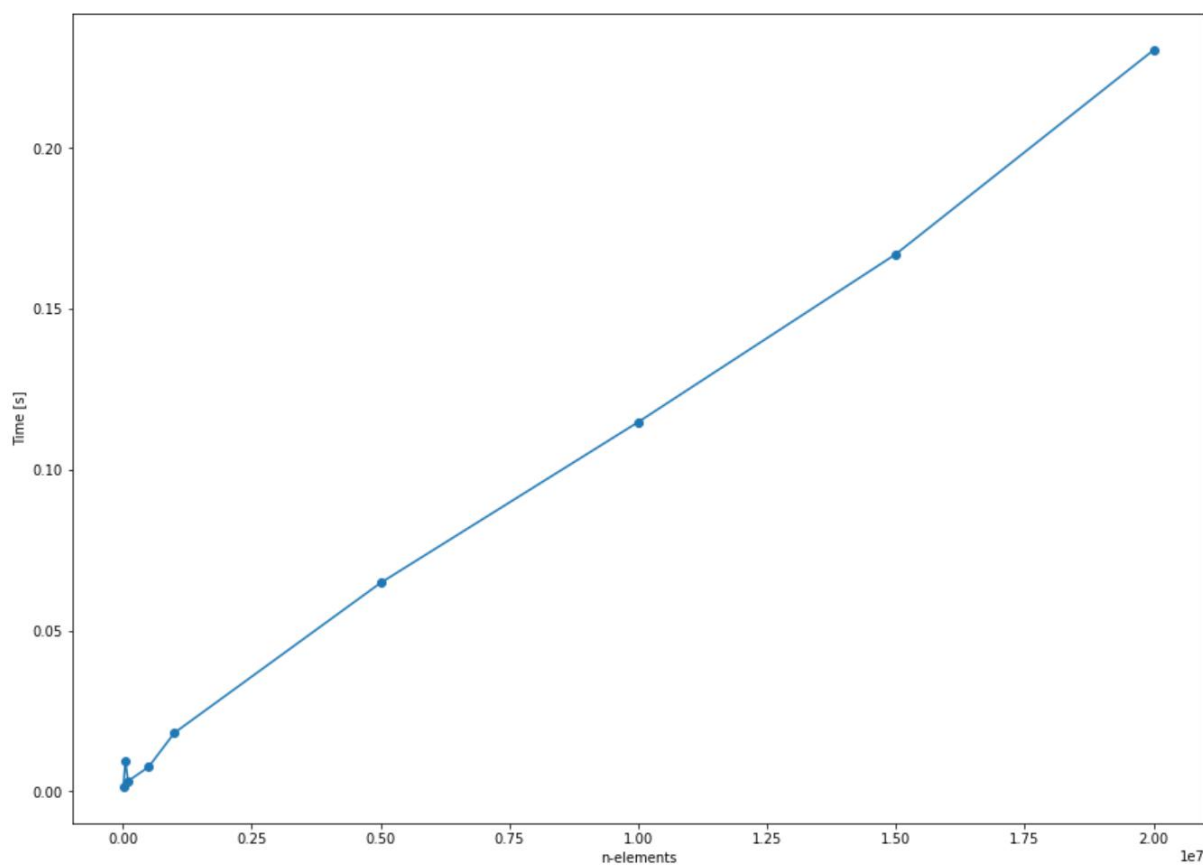
```
            curSeqLen = 0
```

```
    i = i + 1
```

Schemat blokowy:



Wykres złożoności czasowej:



Testy:

Ciąg: -10, 5, 8, 1, -4, -4, 10, 3, -1, 1,

Podciąg no. 0: 8 1 -4 Podciąg no. 1: 10 3 -1

Ciąg: 0, 0, 0, 0, 0,

All elements are equal or passed vector is empty!

Ciąg: 1, 1, 1, 1, 1,

All elements are equal or passed vector is empty!

Ciąg: 895, 883, 456, 863, 208, 103, 127, 146, 334, 550,

Podciąg no. 0: 883 456 863 Podciąg no. 1: 863 208 103

Ciąg: 989, 98, 645, 522, 74, 135, 305, 219, 902, 33, 289, 382, 320, 1, 405,

Podciąg no. 0: 645 522 74 Podciąg no. 1: 382 320 1

Ciąg: 176, 745, 77, 241, 651, 487, 948, 146, 708, 446, 148, 52, 578, 893, 825, 637, 276, 200, 57, 230,

Podciąg no. 0: 893 825 637 276 200 57

Ciąg: 895, 883, 456, 863, 208, 103, 127, 146, 334, 550,

Podciąg no. 0: 883 456 863 Podciąg no. 1: 863 208 103

Ciąg: 989, 98, 645, 522, 74, 135, 305, 219, 902, 33, 289, 382, 320, 1, 405,

Podciąg no. 0: 645 522 74 Podciąg no. 1: 382 320 1

Kod programu *

Wyniki przeprowadzonych testów *

*-link do repozytorium