

NAJVEČJA ABSOLUTNA RAZLIKA MED NAJBLIŽJIM LEVIM IN DESNIM MANJŠIM ELEMENTOM

Avtor : Hana Kranjec Kelbel, Praktična matematika FMF

Predstavitev problema

Imamo tabelo pozitivnih celih števil . Med elementi v tabeli iščemo največjo absolutno razliko med najbližjim levim in desnim manjšim elementom.

Vsakemu elementu v tabeli poiščemo najbližje manjše število na desni in levi strani. Nato primerjamo razdalje med njimi ter poiščemo največjo absolutno razliko.

- **Kaj je absolutna razlika med najbližjim levim in desnim elementom**

Primer : [2,4,5,1]

Zanima nas najbližje manjše levo in desno število elementa 4 iz podane tabele.

- Najbližje manjše levo število je 2.
- Najbližje manjše desno število je 1.



Njuna absolutna razlika je 1.

Na način kot smo opisali v zgornjem primeru bomo poiskali absolutno razliko med najbližjim manjšim levim in desnim elementom za vsak element iz podane tabele.

Prvi element v tabeli nima najbližjega levega elementa. V takem primeru rečemo, da najbližji levi element je 0.

Enako velja za zadnji element v tabeli. Zadnji element nima najbližjega desnega elementa. Tudi njemu določimo najbližji desni element kar 0.

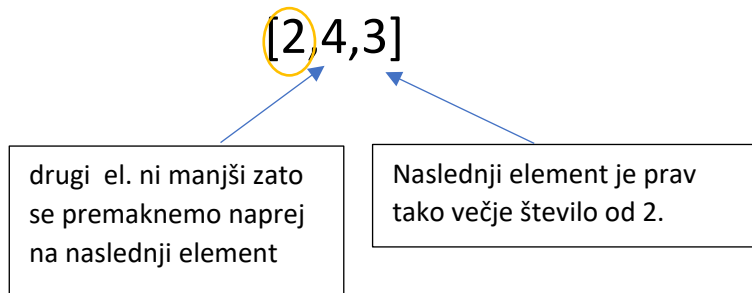
- **Kaj je najbližji desni manjši element?**

Imamo neko zaporedje števil, sedaj nas zanima katero število na njegovi desni strani je najbližje manjše število. Oziroma katero manjše število med elementi zagledamo prvo na desni strani.

Poglejmo si na primeru **[2,4,3]**

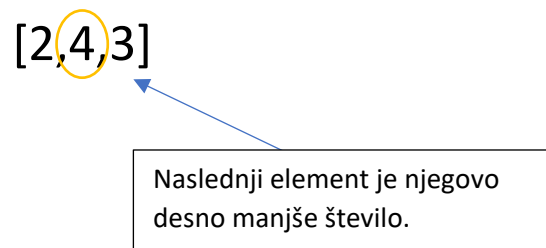
Za vsak element v tabeli bomo poiskali njegovo najbližje desno manjše število. Pomikamo se od leve proti desni strani.

- Prvi element v tabeli je število 2. Poiščemo desni manjši element od števila 2.



Ker nismo našli desnega manjšega elementa, pravimo da je število 0 desni manjši element števila 2.

- Drugi element v tabeli je število 4.



Drugemu elementu v tabeli smo našli desni manjši element in to je število 3.

- Zadnji element v tabeli je število 3. Ker je to zadnji element, zato nima desnega elementa. V takem primeru kar rečemo, da je njegovo desni manjši element 0.

[2,4,3]

- **Kaj je najbližji levi manjši element?**

Imamo neko zaporedje števil in nas zanima katero število na levi strani je najbližje manjše število.

Oziroma katero manjše število med elementi zagledamo prvo na levi strani.

Poglejmo si na primeru $[2,4,3]$

Za vsak element v tabeli bomo poiskali njegovo najbližje levo manjše število.

- Prvi element v tabeli je število 2. Prvi element na levi strani nima nobenega števila. Zato pravimo, da je njegovo najbližje levo število 0.

$[2,4,3]$

Premaknemo se na naslednji element. Premikamo se iz leve proti desni. Najbližje leve elemente pa iščemo od desne proti levi strani.

- Drugi element je število 4

$[2,4,3]$

Pogledamo njegov levi element.
Prvi levi element je manjši.

Našli smo najbližje levo manjše število, zato se premaknemo najprej na naslednji element.

- Naslednji element je število 3.

$[2,4,3]$

Našli smo
najbližje
manjše levo
število

Večji element

Sedaj, ko vemo kaj je najbližje manjše levo ali desno število. Se vrnimo k našem problemu. Iščemo največjo absolutno razliko med najbližjim levim in desnim elementom.

Ideja

Zastavljeno vprašanje lahko rešimo na več načinov. Ogledali si bomo reševanje s pomočjo tabel in podatkovne strukture Sklad. Uporabljali bomo programski jezik Python in v naprej pripravljen razred Sklad, katerega smo definirali na predavanjih.

- Reševanje s tabelo

Podano imamo tabelo števil.

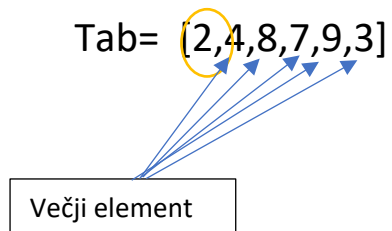
Za vsako število iz tabele poiščemo najbližje desno število in ga zapišemo v novo tabelo.

Nato poiščemo levo najbližje število za vsako število iz tabele. Prav tako najbližje število zapišemo v novo tabelo.

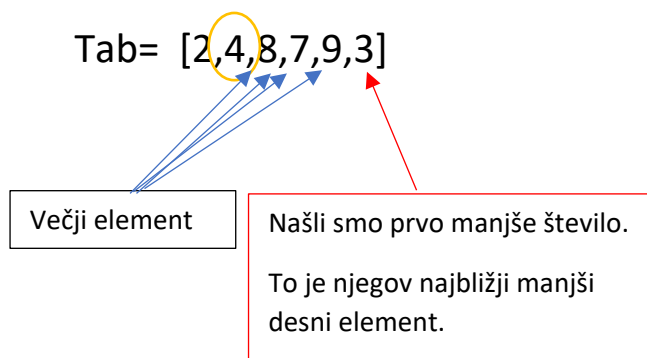
Ker pa iščemo absolutno razliko med najbližjim levim in desnim elementom. Izračunamo razlike med isto ležečimi elementi iz obeh tabel. Izmed teh razlik poiščemo največjo absolutno razliko in to je naša iskana največja absolutna razlika med najbližjim levim in desnim elementom.

Primer [2,4,8,7,9,3]

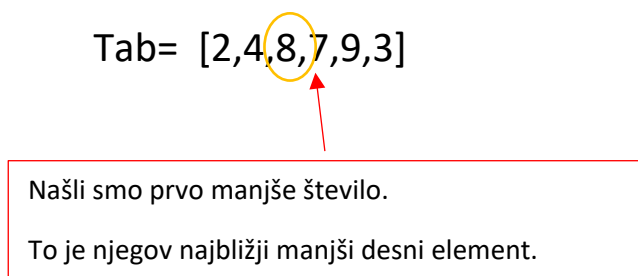
1. Za vsak element iz tabele poiščemo prvo desno manjše število. Dodamo ga na isto ležeče mesto v novi tabeli.



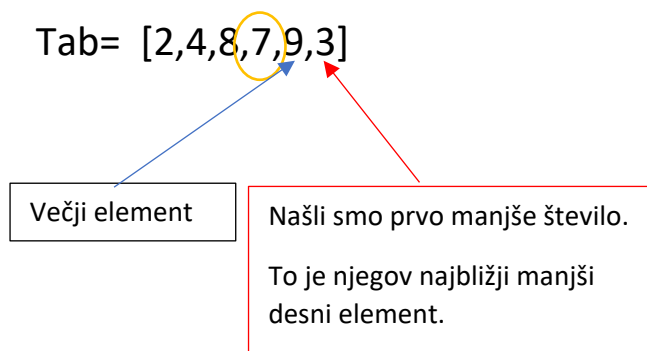
desni = [0]



desni = [0,3]



desni = [0,3,7]



desni = [0,3,7,3]

Tab= [2,4,8,7,9,3]

desni = [0,3,7,3,3]

Našli smo prvo manjše število.

To je njegov najbližji manjši desni element.

Tab= [2,4,8,7,9,3]

desni= [0,3,7,3,3,0]

Zadnji element nima desnega števila, zato je njegovo desno najbližje število 0.

2. Za vsak element iz tabele poiščemo prvo levo manjše število. Dodamo ga na isto ležeče mesto v novi tabeli. Premikamo se od trenutnega elementa proti začetku tabele.

Tab= [2,4,8,7,9,3]

levi = [0]

Prvi element nima levega najbližjega elementa, zato je njegovo levo najbližje število 0.

Tab= [2,4,8,7,9,3]

levi = [0,2]

Našli smo prvo manjše število.

To je njegov najbližji manjši levi element.

Tab= [2,4,8,7,9,3]

levi = [0,2,4]

Našli smo prvo manjše število.

To je njegov najbližji manjši levi element.

Tab= [2,4,8,7,9,3]

levi = [0,2,4,4]

Našli smo prvo
manjše število.

To je njegov
najbližji manjši
levi element.

Večji element

Tab= [2,4,8,7,9,3]

levi = [0,2,4,4,7]

Našli smo prvo manjše število.

To je njegov najbližji manjši levi element.

Tab= [2,4,8,7,9,3]

levi = [0,2,4,4,7,2]

Našli smo prvo
manjše število.

To je njegov
najbližji manjši
levi element.

Večji element

3. Poiščemo razlike med isto ležečimi element iz tabele levih najbližjih in desnih najbližjih elementov. Ena izmed možnosti je da razlike med isto ležečimi elementi iz obeh tabel zapišemo v novo tabelo.

desni= [0,3,7,3,3,0]

Izračunamo
razliko med
isto ležečimi
števili tabeli.

levi = [0,2,4,4,7,2]

razlike = [0,1,3,1,4,2]

4. Poiščemo največjo absolutno razliko med najbližjim levim in desnim elementom.
(Rešitev je maksimum iz tabele razlik)

razlike = [0,1,3,1,4,2] Rešitev našega problema je 4.

- Reševanje s sklado

Podano imamo tabelo števil. Števila iz tabele bomo prenašali v sklad.

Začnemo z iskanjem desnih najbližjih manjših elementov. To storimo tako, da elemente iz seznama prenesemo v sklad. Ker pri skladu uporabimo metodo LIFO, začnemo z zadnjim elementom iz tabele. Nato se premikamo proti začetku tabele oz. dodajmo elemente iz tabele v sklad od desnega proti levemu elementu.

Odstranjujemo vrhnji element iz sklada toliko časa dokler ne zagledamo manjši element od obravnavanega števila. Pri odstranjevanju vrhnjih elementov iz sklada je prvi manjši prav tako naš iskani najbližji desni element.

Poglejmo si na primeru [2,4,8,7,9,3]

Tab = [2,4,8,7,9,3]
desni = [0]



Iščemo desne najbližje manjše elemente. Začnemo z zadnjim elementom. Ustvarimo prazen sklad.

Ker smo začeli pri zadnjem elementu, vemo, da zadnji element nima desnega elementa. Zato zapišemo v tabelo desnih najbližjih najmanjših elementov vrednost 0.

Tab = [2,4,8,7,9,3]
Desni = [0,3]



Premaknemo se za en element v levo po tabeli.

V sklad dodamo zadnji element.

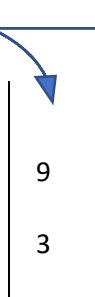
Pogledamo vrhnji element v skladu.

Primerjamo obravnavani element in vrhnji element iz sklada. Vrhnji element je manjši – našli smo desni najbližji manjši element.

Dodamo ga v tabelo rezultatov.

Tab = [2,4,8,7,9,3]

Desni = [0,3]



Premaknemo se za en element v levo po tabeli.

V sklad dodamo naslednji element.

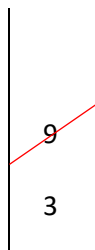
(V sklad dodajamo od desne -> levi)

Pogledamo vrhnji element v skladu.

Vrhnji element je večji od obravnavanega, zato ga odstranimo.

Tab = [2,4,8,7,9,3]

Desni = [0,3,3]

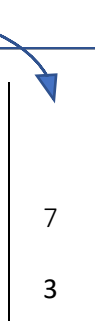


Ponovno primerjamo obravnavani element in vrhnji element iz sklada. Vrhnji element je manjši – našli smo desni najbližji manjši element.

Dodamo ga v tabelo rezultatov.

Tab = [2,4,8,7,9,3]

Desni = [0,3,3,7]



Premaknemo se za en element v levo po tabeli.

V sklad dodamo naslednji element.

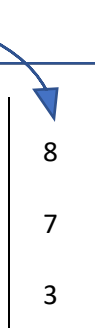
Pogledamo vrhnji element v skladu.

Vrhnji element je manjši od obravnavanega.

Dodamo ga v tabelo rezultatov.

Tab = [2,4,8,7,9,3]

Desni = [0,3,3,7]



Premaknemo se za en element v levo po tabeli.

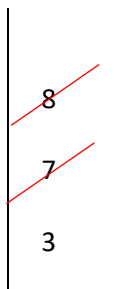
V sklad dodamo naslednji element.

Pogledamo vrhnji element v skladu.

Vrhnji element je večji od obravnavanega, zato ga odstranimo.

Tab = [2,4,8,7,9,3]

Desni = [0,3,3,7,3]



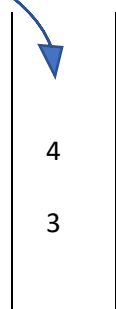
Ponovno pogledamo vrhnji element v skladu.

Vrhnji element je večji od obravnavanega, zato tudi ta element odstranimo.

Spet pogledamo vrhnji element, ker je manjši ga dodamo v tabelo rezultatov.

Tab = [2,4,8,7,9,3]

Desni = [0,3,3,7,3]



Premaknemo se za en element v levo po tabeli.

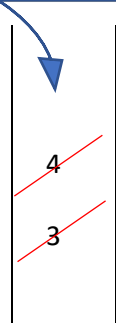
V sklad dodamo naslednji element.

Pogledamo vrhnji element v skladu.

Vrhnji element je večji od obravnavanega, zato ga odstranimo.

Tab = [2,4,8,7,9,3]

Desni = [0,3,3,7,3,0]



Pogledamo naslednji vrhnji element.

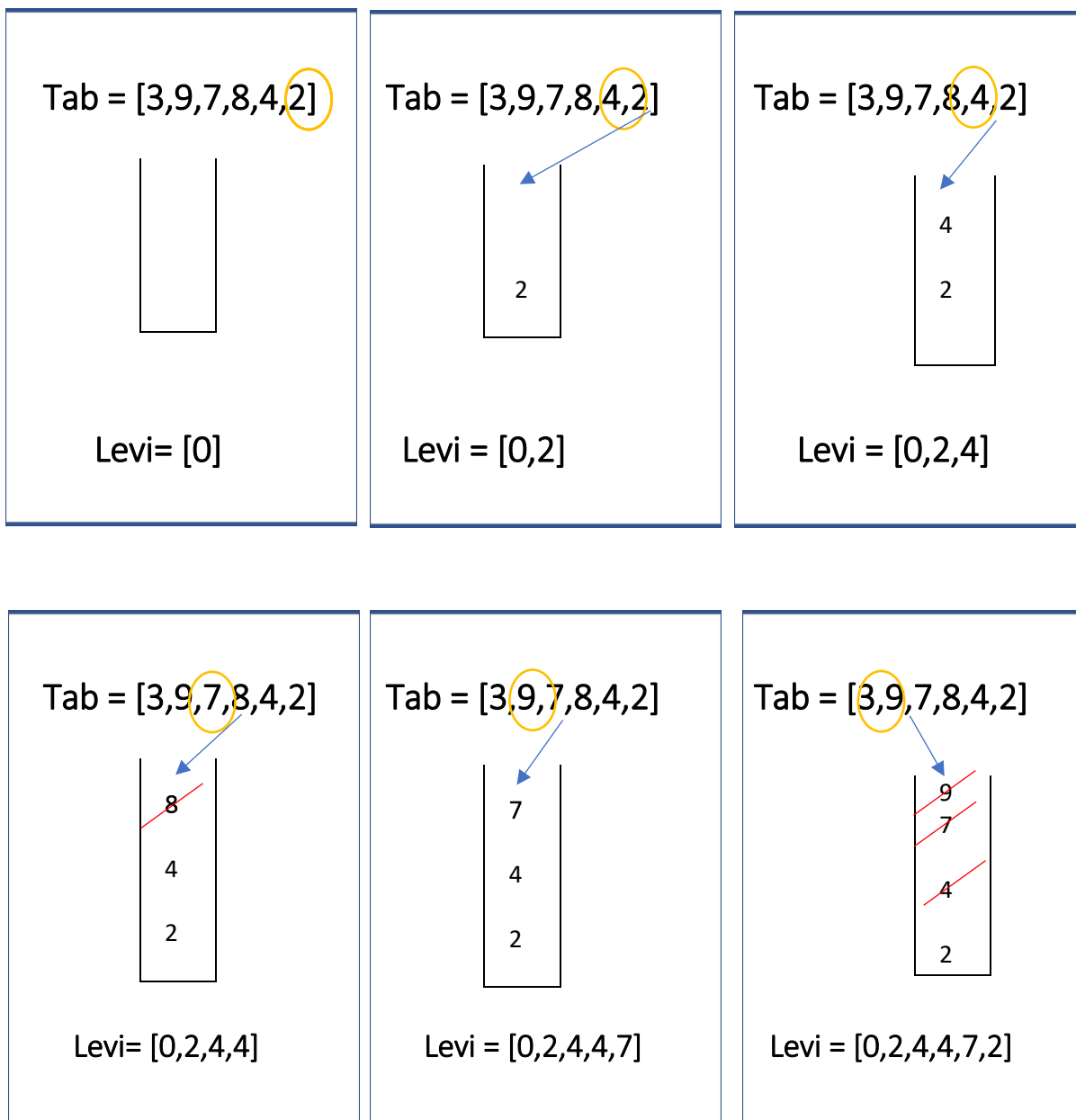
Tudi ta je večji od obravnavanega, zato tudi ta element odstranimo.

Sedaj je sklad prazen. To pomeni, da obravnavani element nima manjšega elementa in zato v tabelo rezultatov dodamo število 0.

Sedaj smo dobili tabelo desnih najbližjih manjših elementov. Ker smo začeli z zadnji elementom in se pomikali proti prvemu elementu iz tabele, moramo tabelo rešitev obrniti, da dobimo pravo rešitev.

Desni = [0,3,7,3,3,0]

Če želimo izračunati absolutne razlike med najbližjimi levimi in desnimi manjšimi elementi. Moramo sedaj poiskati še najbližje leve elemente. To bomo storili na enak način. Le, da bomo tabelo na začetku obrnili.

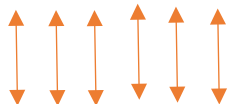


Sedaj smo dobili tabelo levih najbližjih manjših elementov.

Levi = [0,2,4,4,7,2]

Izračunamo še absolutne razlike med isto ležečimi elementi v levi in desni tabeli. Izmed vseh razlik poiščemo največjo in je naša iskala rešitev.

desni= [0,3,7,3,3,0]



levi = [0,2,4,4,7,2]

Izračunamo
razliko med
isto ležečimi
števíli tabeli.

razlike = [3,1,3,1,4,2]

Rešitev

- Rešitev z uporabo tabele

```
def maxRazlika(tab):  
    '''  
    Funkcija ustvari dve tabeli iz desnih in levih najmanjših najbližjih elementov.  
    Izračuna absolutne razlike med elementi iz obeh tabel in vrne največjo razliko.  
    '''  
    if tab == [] or len(tab)==1:  
        return False  
  
    desni=[] # v to tabelo shranimo desne najbližje in najmanjše vrednosti  
    for i in range(0, len(tab)): #za vsak element iz tabele bomo poiskali najbližjega  
        next = 0 # če manjšega ni  
        for j in range(i + 1, len(tab)): #vsak element desno od izbranega preverimo ali je manjši  
            #začnemo z naslednjim el.  
            if tab[i] > tab[j]: #desni el. je manjši od trenutnega  
                next = tab[j] #če smo ga našli ga shranimo v tabelo  
                break  
        desni.append(next)  
  
    levi = []  
    obratniTab=tab[::-1] #obrnemo tabelo in ponovno iščemo desne najbližje tako dobim leve najbližje  
    for i in range(0, len(obratniTab)):  
        next = 0  
        for j in range(i + 1, len(obratniTab)):  
            if obratniTab[i] > obratniTab[j]:  
                next = obratniTab[j]  
                break  
        levi.append(next) # na koncu še enkrat obrnjemo tabelo, da dobimo res leve najbližje  
  
    razlike=[]  
    for l,d in zip(desni, levi[::-1]): #med vsemi elementi iz novih tabel izračunamo absolutne razlike  
        razlike.append(abs(l-d))  
    print(" Tabela desnih najbližjih elementov: ",desni)  
    print("Tabela levih najbližjih elementov: ",levi[::-1])  
    print("Tabela absolutnih razlik med levimi in desnimi najbližjimi elementi: ",razlike)  
    print("Največja absolutna razlika med levimi in desnimi najbližjimi elementi: ")  
    return max(razlike) # vrnemo največjo razliko
```

```
>>> maxRazlika([2,1,4,3])  
Tabela desnih najbližjih elementov: [1, 0, 3, 0]  
Tabela levih najbližjih elementov: [0, 0, 1, 1]  
Tabela absolutnih razlik med levimi in desnimi najbližjimi elementi: [1, 0, 2, 1]  
Največja absolutna razlika med levimi in desnimi najbližjimi elementi:  
2  
...
```

- Rešitev z uporabo razreda Sklad

```
def naslednji_najmanjsi_el(tab):
    '''
    Funkcija s pomočjo sklada iz podanega seznama vrne seznam
    najbližjih in najmanjših elementov.
    '''
    if tab == [] or len(tab)==1:
        return False

    sklad = Sklad()
    rez = []
    # s for zanko pogledamo vse elemente v tabeli
    for i in range(len(tab)-1, -1, -1): # desna -> leva
        if sklad.prazen(): #če je sklad prazen, ni naslednjega min.
            rez.append(0)
            sklad.vstavi(tab[i]) #v sklad vstavimo el iz tabele
        elif not sklad.prazen(): # v primeru, da sklad ni prazen
            while(not sklad.prazen() and tab[i] < sklad.vrh()): # preverjamo el. v skladu dokler ga ne izpraznimo in
                # dokler so vrhnji el iz sklada večji od obravnavanega el.
                #odstranjujemo vrhnje oz. večje elemente
                sklad.odstrani()
            if sklad.prazen(): # če nismo našli manjšega
                rez.append(0)
            else:
                rez.append(sklad.vrh()) # našli smo manjši el.
                sklad.vstavi(tab[i])

    return rez
```

```
def maxEl(tabela):
    '''
    Funkcija vrne največji element med absolutnimi razlikami iz dveh tabel.
    '''
    desni = naslednji_najmanjsi_el(tabela) #seznam desnih najmanjših el
    tabD = desni[::-1] #obreno tabelo rešitev, ker smo pregledovali el od desne->levi

    #leve najmanjše el. dobimo enako kot desne, le da smo obrnili tabelo
    tabL = naslednji_najmanjsi_el(tabela[::-1]) #tabela levih najmanjših el

    rezultat = []
    for l,d in zip(tabL,tabD): #absolutne razlike med el. iz dveh tab.
        rezultat.append(abs(l-d))
    return rezultat, max(rezultat)
```

```
>>> maxEl([2,4,8,7,9,3])
[0, 3, 7, 3, 3, 0]
[0, 2, 4, 4, 7, 2]
([0, 1, 3, 1, 4, 2], 4)
>>>
```

Viri

1. Vprašanje poišči največjo razliko med levim in desnim manjšim elementom : <https://www.geeksforgeeks.org/find-maximum-difference-between-nearest-left-and-right-smaller-elements/> [Dostop : 13.12.2021]
2. Naslednji manjši element v seznamu : <https://www.geeksforgeeks.org/next-smaller-element/> [Dostop : 13.12.2021]
3. Razred Sklad <https://gist.github.com/gregaSaksida/1f7559aca55c2997e40f8074b33ecf89> [Dostop : 13.12.2021]