



Modeli peščenih stolpcev

Python

Python urejevalnik se nahaja na strani:

<https://www.w3resource.com/python-exercises/python-basic-exercises.php#EDITOR>

Razlago si pogledajte na naslovu:

<https://www.youtube.com/watch?v=1MtEUErz7Gg>

Peščeni stolpci

Vzemimo $n \times n$ celic, razporejenih v n vrstic po n celic. Na posamezne celice polagamo zrnca peska. Na vsaki celici je mesto za največ eno zrnce. Lahko pa zrnca polagamo eno vrh drugega, vendar pa stolpec ne more biti višji od treh zrn. Višji stolpec se podre in odda zrnca sosedom.

Definirajmo matematično: *matrika peščenih stolpcev* je matrika s $n \times n$ celicami, v katerih so zapisana števila od 0 do 3.

Seštevanje matrik peščenih stolpcev bomo definirali po naravni poti, kot združevanje zrn iste vrste celic obeh sumandov. Vsoto matrik peščenih stolpcev bomo imenovali kar *peščeno vsoto*.

Torej bomo vsoto dveh matrik izračunali tako, da ju položimo eno vrh druge in združimo zrnca celic, ki se prekrivajo. Vendar pa to ni vse. Rekli smo, da se stolpci, višji od treh zrn, začnejo podirati.

Stolpci se podirajo tako, da se znižajo za 4 zrnca, pri tem pa oddajo vsaki od sosednjih celic, gornji, spodnji, levi in desni, po eno zrnce. Tista zrnca, ki padejo čez rob, so izgubljena. S tem postopkom prečesavamo matriko toliko časa, da je število zrn v vseh celicah manjše ali enako tri. Tako dobljena matrika predstavlja peščeno vsoto matrik peščenih stolpcev.

Primer peščene vsote dveh matrik peščenih stolpcev:

$$\begin{array}{|c|c|c|} \hline 3 & 3 & 3 \\ \hline 3 & 3 & 3 \\ \hline 3 & 3 & 3 \\ \hline \end{array} + \begin{array}{|c|c|c|} \hline 3 & 3 & 3 \\ \hline 3 & 1 & 3 \\ \hline 3 & 3 & 3 \\ \hline \end{array} = \begin{array}{|c|c|c|} \hline 2 & 1 & 2 \\ \hline 1 & 0 & 1 \\ \hline 2 & 1 & 2 \\ \hline \end{array}$$

Definicija problema

Napišite program, ki bo poiskal peščeno vsoto dveh matrik reda $n \times n$, katerih elementi so $0 \leq a_{ij} \leq 3$

Postopek računanja peščene vsote matrik gornjega primera. Vrednosti popravljamo po vrsticah, začnemo v zgornjem levem kotu in končamo v spodnjem desnem kotu.

6	6	6	4	5	4	2	4	2
6	4	6	5	4	5	4	4	4
6	6	6	4	5	4	2	4	2
4	1	4	0	4	1	2	1	2
1	4	2	4	2	0	1	0	1
4	2	0	1	0	2	2	1	2

Primera

Vhod

1. $[[3, 3, 3], [3, 3, 3], [3, 3, 3]] + [[3, 3, 3], [3, 1, 3], [3, 3, 3]]$
2. $[[3, 2, 3], [2, 3, 2], [3, 1, 3]] + [[2, 1, 2], [1, 0, 1], [2, 1, 2]]$

Izhod

1. $[[2, 1, 2], [1, 0, 1], [2, 1, 2]]$
2. $[[3, 2, 3], [2, 3, 2], [3, 1, 3]]$

Rešitev

```
#!/usr/bin/env python3
import math

def add_sp(a, b):
    x = list(map(lambda x, y: x + y, a, b))
    ch = True
    n = len(x)
    m = int(math.sqrt(len(x)))
    while ch:
        ch = False
        for k in range(n):
            if x[k] > 3:
                ch = True
                x[k] -= 4
                i = k // m
                j = k % m
                if i > 0:
                    x[k - m] += 1
                if i < m - 1:
                    x[k + m] += 1
                if j > 0:
                    x[k - 1] += 1
                if j < m - 1:
                    x[k + 1] += 1
    return (x)

if __name__ == "__main__":
    a = [3, 3, 3, 3, 3, 3, 3, 3, 3]
    b = [3, 3, 3, 3, 1, 3, 3, 3, 3]
    print(add_sp(a,b))
    a = [3,2,3,2,3,2,3,1,3]
    b = [2,1,2,1,0,1,2,1,2]
    print(add_sp(a, b))
```