

Računalništvo (fizika): 2. izpit

11. 5. 2017

Naloge rešujte na strežniku Tomo, <https://www.projekt-tomo.si>.

Brskanje po spletni učilnici, nalogah na strežniku Tomo, video zapiskih predavanj in *uradni* dokumentaciji za Python *je dovoljeno*. Brskanje po stackoverflow.com, rosettacode.org in ostalih spletnih straneh, ki vsebujejo rešitve nalog in nasvete o tem, kako se rešuje naloge, *ni dovoljeno*. Komunikacija s komerkoli, razen z asistentom, *ni dovoljena*.

Čas reševanja je 180 minut. Doseženih 100 točk šteje za maksimalno oceno. Veliko uspeha!

1. naloga (30 točk)

Podan je seznam n celih števil $s = [x_0, \dots, x_{n-1}]$. Zanima nas, kako bi seznam razdelili na dva dela, levega in desnega, da dobimo največji možni zmnožek vsot obeh delov. Na primer, seznam $[-4, 5, -3, 7, -2, 2]$ bi lahko razrezali na sedem načinov in dobili zmnožke:

$$\begin{aligned}0 \cdot (-4 + 5 - 3 + 7 - 2 + 2) &= 0 \\(-4) \cdot (5 - 3 + 7 - 2 + 2) &= -36 \\(-4 + 5) \cdot (-3 + 7 - 2 + 2) &= 4 \\(-4 + 5 - 3) \cdot (7 - 2 + 2) &= -14 \\(-4 + 5 - 3 + 7) \cdot (-2 + 2) &= 0 \\(-4 + 5 - 3 + 7 - 2) \cdot (2) &= 6 \\(-4 + 5 - 3 + 7 - 2 + 2) \cdot 0 &= 0\end{aligned}$$

Kot vidimo, dobimo največji zmnožek 6, če seznam razdelimo na $[-4, 5, -3, 7, -2]$ in $[2]$.

a) (25 točk) Sestavite funkcijo `najvecja_vrednost(s)`, ki sprejme seznam s in izračuna največjo možno vrednost zmnožka vsot levega in desnega dela.

b) (5 točk) Funkcija `najvecja_vrednost` naj deluje v času $O(n)$. Pozor: seštevaje elementov v seznamu dolžine k zahteva k računskih korakov, tudi če uporabite vgrajeno funkcijo `sum`.

2. naloga (25 točk)

Aljaž ima funkcijo f , ki slika naravna števila v cela števila. Definira novo funkcijo s , ki računa delne vsote f :

$$s(k) = f(1) + f(2) + \dots + f(k).$$

Za dano število n išče tak $1 \leq k \leq n$, da ima $s(k)$ najmanjšo vrednost.

a) (10 točk) Sestavite funkcijo `delne_vsote(f)`, ki sprejme funkcijo f in vrne pripadajočo funkcijo s , ki računa delne vsote f .

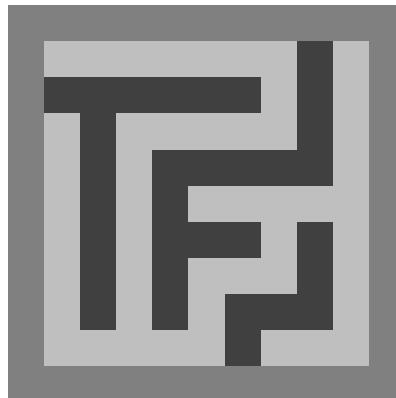
b) (15 točk) Sestavite funkcijo `minimalna_delna_vsota(f, n)`, ki sprejme f in n ter vrne tisti $1 \leq k \leq n$, pri katerem $s(k)$ zasede minimum. Če je minimumov več, naj funkcija vrne najmanjši k , pri katerem je minimum zaseden.

3. naloga (25 točk)

Študent fizike se je izgubil v labirintu. Pomagaj mu poiskati pot do izhoda.

Labirint predstavimo z matriko A velikosti $n \times m$. Z ničlami v matriki predstavimo prazna polja in z enicami stene. Na primer, matrika na levi predstavlja labirint na desni (zunanje stene niso del matrike):

0	0	0	0	0	0	0	1	0
1	1	1	1	1	1	0	1	0
0	1	0	0	0	0	0	1	0
0	1	0	1	1	1	1	1	0
0	1	0	1	0	0	0	0	0
0	1	0	1	1	1	0	1	0
0	1	0	1	0	0	0	1	0
0	1	0	1	0	1	1	1	0
0	0	0	0	0	1	0	0	0



Študent vedno začne v levem zgornjem kotu, na koordinatah $(0,0)$, izhod pa je v desnem spodnjem kotu, na koordinatah $(n-1, m-1)$.

Sestavite funkcijo `pot(a)`, ki sprejme matriko `a` z opisom labirinta in vrne pot od zgornjega levega do spodnjega desnega polja. Pot je podana s seznamom $[(i_0, j_0), (i_1, j_1), \dots, (i_k, j_k)]$ koordinat polj, ki jih mora študent prehoditi. Koordinate (i, j) označujejo polje v i -ti vrstici in j -tem stolpcu. Če rešitve ni, naj funkcija vrne `None`. Če je rešitev več, naj vrne katerokoli.

Namig: nalogo lahko rešite z rekurzivno funkcijo, ki sprejme do sedaj prehojeno pot `p` in jo poskuša rekurzivno razširiti na enega od sosedov zadnjega polja v `p`. Pri tem se izogiba že prehojenim poljem, da se ne bi ujela v cikel.

4. naloga (30 točk)

Sestavite funkcijo `odstrani_ponovljene(s)`, ki sprejme tabelo `s` in vrne novo tabelo, ki ima enake elemente kot `s`, le da se ti ne ponavljajo. Na primer

```
>>> odstrani_ponovljene([2, 1, 3, 1, 0, 0, 4])
[4, 0, 1, 2, 3]
```

Vrstni red elementov v vrnjeni tabeli ni pomemben. Rešitev naj deluje v času $O(n \cdot \log n)$.

Navodilo: nalogo rešite "na roke", se pravi da implementirate ustrezeni algoritem. Rešitev, ki uporablja Pythonove vgrajene funkcije, denimo `list(set(s))`, ni dopustna (poleg tega ni razvidno, kakšna je v tem primeru časovna zahtevnost).