

Računalništvo (fizika): 2. izpit

10. junij 2019

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 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 (25 točk)

a) (10 točk) V danem slovarju so ključi cela števila. Sestavite funkcijo `naj_vrednost(s)`, ki sprejme tak slovar `s`, in vrne vrednost, ki pripada največjemu ključu. Če je slovar prazen, naj funkcija vrne `None`.

Primer uporabe:

```
>>> naj_vrednost({ 5 : "foo", 3 : "bar", 2 : "qux"})
'foo'
```

b) (15 točk) V danem slovarju `s` so vrednosti cela števila. Dana je funkcija `p`, ki slika ključ v Boolove vrednosti. Zanima nas vsota vrednosti, ki pripadajo tistim ključem, za katere `p` vrne `True`. Sestavite funkcijo `filter_vsota(s, p)`, ki to izračuna.

Primeri uporabe:

```
>>> filter_vsota({'Ana' : 1, 'Bine' : 3, 'Mojca' : 10, 'Marija' : 15},
                 (lambda ime: len(ime) % 2 == 0))
18
>>> filter_vsota({}, (lambda x : True))
0
>>> filter_vsota({'foo' : 42, 'bar' : 23}, (lambda x : False))
0
```

2. naloga (35 točk)

Na planetu Resurgam živita dve vrsti nezemljanov, Bynari in Grebuloni. Ti dve vrsti nezemljanov sta še posebej zanimivi, saj se poleg razmnoževanja znotraj vrste, lahko razmnožujeta tudi ena z drugo. Obe vrsti opišemo z dvema atributoma, velikostjo ter odtenkom barve; obe vrednosti merimo kot števili.

Pri razmnoževanju Bynarov in Grebulonov se oba atributa določita po natanko določeni formuli, ki pa je odvisna od vloge posameznikov – pri vsakem razmnoževanju vsak posameznik namreč zavzame natanko eno vlogo. Za lažjo predstavo bomo vlogi imenovali mati in oče. Atributa matere in očeta teda predstavimo kot vektorja $\vec{m} = [velikost_{mati}, odtenek_{mati}]$ in $\vec{o} = [velikost_{oce}, odtenek_{oce}]$.

Kadar je mati Bynar, se atributi naraščaja izračunajo kot

$$velikost = \vec{m} \begin{bmatrix} -1.1 & -0.2 \\ -0.8 & 1.1 \end{bmatrix} \vec{o}^\top \text{ in } odtenek = \vec{m} \begin{bmatrix} -1.8 & 0.5 \\ 1.0 & -0.9 \end{bmatrix} \vec{o}^\top,$$

kadar pa je mati Grebulon pa kot

$$velikost = \vec{m} \begin{bmatrix} 0.7 & -0.2 \\ 1.2 & 0.8 \end{bmatrix} \vec{o}^\top \text{ in } odtenek = \vec{m} \begin{bmatrix} -1.8 & 1.4 \\ 0.9 & 0.1 \end{bmatrix} \vec{o}^\top.$$

Vrsta naraščaja pa je določena po sledečem pravilu: če sta tako mati kot oče iste vrste, je tudi naraščaj iste vrste, sicer pa vrsto določa produkt velikosti in odtenka naraščaja. Kadar je produkt večji ali enak 0, je naraščaj iste vrste kot mati, sicer pa iste vrste kot oče.

a) (10 točk) Sestavite razreda `Bynar` in `Grebulon`, ki preko konstruktorja prejmeta in shranita vrednosti velikost in odtenka. Poleg konstruktorja razredoma dodajte še metodi `sem_bynar` in `sem_grebulon`, ki vrneta ustrezni resničnostni vrednosti za vsak razred (npr. `sem_bynar` vrne `True` za objekte razreda `Bynar` in `False` za objekte razreda `Grebulon`).

b) (10 točk) Razredoma dodajte metodo `__repr__` za predstavitev objekta in metodo `__eq__` za primerjanje enakosti. Pozor: pri preverjanju enakosti preverite enakost velikosti in odtenka, ne pozabite pa tudi vrsto nezemljanov.

c) (15 točk) Razmnoževanje bomo v Python prevedli kot operacijo seštevnja. Razredoma dodajte še metodo `__add__`, ki po zgornjih navodilih izračuna nov objekt, ki predstavlja naraščaj.

3. naloga (30 točk)

Slovenski matematik Ivo Lah je odkril števila, ki se imenujejo po njem. Za nenegativni celi števili n in k je Lahovo število $L(n, k)$ določeno s predpisom:

$$L(n, k) = \begin{cases} 0 & \text{če je } k = 0 \text{ ali } k > n, \\ 1 & \text{če je } k = n = 1, \\ (n + k) \cdot L(n, k) + L(n, k - 1) & \text{sicer} \end{cases}$$

Sestavite funkcijo `lah(n, k)`, ki vrne vrednost $L(n, k)$. Funkcija naj deluje učinkovito tudi za večje vrednosti n in k . Primeri uporabe:

```
>>> lah(8, 2)
141120
>>> lah(42, 23)
13296933538639725143138721180549120000000
```

4. naloga (30 točk)

Aljaž se dolgočasi na pisnem izpitu, zato preizkuša, katera števila lahko dobi s seštevanjem in množenjem natanko n enic. Na primer, če je $n = 5$, lahko dobi naslednja števila:

$$\begin{aligned}1 &= 1 \cdot 1 \cdot 1 \cdot 1 \cdot 1 \\2 &= 1 \cdot 1 \cdot 1 \cdot 1 + 1 \\3 &= 1 \cdot 1 \cdot 1 + 1 + 1 \\4 &= 1 \cdot 1 + 1 + 1 + 1 \\5 &= 1 + 1 + 1 + 1 + 1 \\6 &= (1 + 1) \cdot (1 + 1 + 1)\end{aligned}$$

Nekatera števila lahko dobi na več načinov, na primer $4 = (1 + 1 \cdot 1) \cdot (1 + 1)$, števila 7 pa s petimi enicami ne more dobiti. Za dani n ga zanima najmanjše število k_n , ki ga *ne* more dobiti na tak način. Na roke je izračunal, da velja:

n	1	2	3	4	5	6
k_n	2	3	4	5	7	10

Sestavite funkcijo `nedosegljivo(n)`, ki vrne najmanjše naravno število, ki ga ne moremo dobiti iz n enic s seštevanjem in množenjem. Primer uporabe:

```
>>> nedosegljivo(12)
47
```