

Računalništvo (fizika): 1. izpit

28. 1. 2016

Naloge rešujte na strežniku Tomo (<https://tomo.fmf.uni-lj.si/>). V sistem se prijavite s svojim uporabniškim računom na omrežju FMF.

Čas reševanja je 150 minut. Doseženih 100 točk šteje za maksimalno oceno. Veliko uspeha!

1. naloga (30 točk)

Študentje fizike se med odmorom igrajo izštevanko (izbrati morajo srečneža, ki bo pobrisal tablo). Postavijo se v krog, začnejo pri enem od študentov in se po vrsti izločajo: “*Ti ostaneš, ti greš ven, ti ostaneš, ti greš ven, ...*” Zmagovalec je tisti, ki ostane zadnji v krogu.

Na primer, če se igra pet študentov, ki jih po vrsti oštevilčimo 0, 1, 2, 3, 4 in začnemo izštevati pri študentu 0, potem gredo ven po vrsti 1, 3, 0 in 4, zmagovalec je študent 2.

V splošnem n študentov v krogu oštevilčimo s števili $0, \dots, n-1$ tako, da študent i stoji med študentoma $(i-1) \% n$ in $(i+1) \% n$, kjer je $x \% y$ ostanek pri deljenju x z y . Izštevati začnemo pri študentu 0.

Sestavite funkcijo `izstevanka(n)`, ki vrne zmagovalca igre z $n \geq 1$ študenti. Zgled:

```
>>> izstevanka(5)
2
```

2. naloga (30 točk)

Dan je razred *Seznam*, s katerim predstavimo povezan seznam.

```
class Seznam:
```

```
    def __init__(self, glava=None, rep=None):
        if rep is None:
            self.prazen = True
        else:
            self.prazen = False
            self.glava = glava
            self.rep = rep

    def __repr__(self):
        if self.prazen:
            return 'Seznam()'
        else:
            return 'Seznam({0}, {1})'.format(self.glava, self.rep)
```

a) (10 točk) Sestavite metodo `dodaj(self,x)`, ki doda x na *konec* seznama `self`. Zgled:

```
>>> s = Seznam(3, Seznam(5, Seznam(1, Seznam())))
>>> s.dodaj(7)
>>> s
Seznam(3, Seznam(5, Seznam(1, Seznam(7, Seznam()))))
```

b) (20 točk) Sestavite metodo `zbrisi(self,x)`, ki odstrani iz seznama `self` vse pojavitve x . Zgled:

```
>>> s.zbrisi(1)
>>> s
Seznam(3, Seznam(5, Seznam(7, Seznam())))
```

3. naloga (25 točk)

Dana je neprazna tabela $a = [a_0, \dots, a_n]$ različnih celih števil. *Lokalni minimum* je tak element a_i , ki je manjši od svojih sosedov (elementa a_0 in a_n imata vsak po enega sosedo, ali nobenega, če je $n = 0$). V tabeli je lahko več lokalnih minimumov, na primer

[0, 9, 2, 5, 6, 1, 4, 8]

ima lokalne minime 0, 2 in 1. Sestavite funkcijo `lokalni_minimum(a)`, ki sprejme tabelo a in vrne enega od lokalnih minimumov. Funkcija naj deluje v času $O(\log n)$. Zgled:

```
>>> lokalni_minimum([0, 9, 2, 5, 6, 1, 4, 8])
1
```

4. naloga (25 točk)

Nedružabni Slovenci zahajajo v bar, v katerem je ob šanku vrsta tapeciranih barskih stolov, oštevilčenih od 1 do n . Prvi gost zasede stol 1, vsak naslednji pa izbere tak stol, da je čim bolj oddaljen od gostov, ki so že v baru. (Želi maksimizirati razdaljo do najbližjega gosta.) Če ima več možnosti, izbere stol z najnižjim zaporednim številom. Če bi se moral nov gost usesti zraven kakega drugega gosta, raje zapusti bar. Če je v baru 8 stolov, se bodo štirje gosti po vrsti usedli takole,

1			3		4		2
---	--	--	---	--	---	--	---

nato pa bo zmanjkalo prostora. Naj bo B_n največje število gostov, ki se lahko usedejo na n stolov, torej $B_1 = 1, B_2 = 1, B_3 = 2, B_4 = 2, B_5 = 3, \dots$

Sestavite funkcijo `bar(n)`, ki učinkovito izračuna vrednost B_n . V komentarju k vaši rešitvi zapišite, kakšno časovno zahtevnost ima vaša rešitev v odvisnosti od n . Zgled:

```
>>> bar(3)
2
>>> bar(8)
4
```