Računalništvo (fizika): 1. izpit

1. 2. 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 (30 točk)

Sestavite generator `moj_range(a,b,k)`, ki deluje tako kot Pythonov vgrajeni generator `range(a,b,k)`, pri čemer *ne smete* uporabiti `range`! (Priporočamo uporabo zanke `while`.)

Natančneje, `moj_range(a,b,k)` vrne člene aritmetičnega zaporedja

$$a, a + k, a + 2k, \ldots,$$

pri čemer ja zadnji člen zaporedja določen takole:

1. če je $k > 0$, zaporedje narašča in generiramo člene, dokler so strogo manjši od b,
2. če je $k < 0$, zaporedje pada in generiramo člene, dokler so strogo večji od b,
3. če je $k = 0$, ne generiramo nobenih členov.

2. naloga (30 točk)

Sestavite razred `Tocka`, s katerim predstavimo točkasto telo v ravnini z dano maso in hitrostjo. Konstruktor naj sprejme parameter x, y, v_x, v_y in m, pri čemer je (x, y) lega telesa v ravnini, (v_x, v_y) je hitrost in m masa. Razredu `Tocka` dodajte naslednje metode:

1. `__repr__(self)`, ki vrne niz, ki predstavlja telo,
2. `energija(self)`, ki vrne kinetično energijo telesa,
3. `trk(self, other)`, ki vrne *novo* točkasto telo, ki ga dobimo, ko telo `self` *neelastično* trči s telesom `other`; metodi ni treba preverjati, ali sta legi teles enaki.

Nato sestavite še funkcijo `tezisce(lst)`, ki sprejme seznam teles in vrne koordinate njihovega težišča. (Funkcijo `tezisce` definirajte zunaj razreda `Tocka`.)

3. naloga (30 točk)

Algoritem "k-means" sprejme dva seznama točk:

1. seznam *voditeljev* $[v_1, \ldots, v_k]$,
2. seznam *primerkov* $[p_1, \ldots, p_n]$.

Primerke razdeli na k množic $S_1, \ldots, S_k$ in sicer postavi primerek p_i v množico S_j, če je vzorcu v_j najbližji voditelj vzorcu p_i:

$$p_i \in S_j \iff d(p_i, v_j) = \min\{d(p_i, v_\ell) \mid 1 \leq \ell \leq k\},$$

pri čemer smo z $d(a, b)$ označili razdaljo med točkama a in b. V tej nalogi predpostavimo, da ležijo točke v ravnini in da so podane s parom koordinat, t.j., $v_i = (x_i, y_i)$ in $p_j = (x'_j, y'_j)$.

Sestavite funkcijo `klasificiraj(voditelji, vzorci)`, ki sprejme seznama voditeljev in vzorcev ter vrne seznam *množic* $[S_1, \ldots, S_k]$, kakor je to opisano zgoraj.

4. naloga (30 točk)

Sestavite funkcijo `drevesa(n)`, ki prešteje, koliko je dvojiških dreves z `n` vozlišči. Funkcija mora delovati učinkovito tudi za veče vrednosti `n`.

Primer uporabe:

```
>>> [drevesa(n) for n in range(10)]
[1, 1, 2, 5, 14, 42, 132, 429, 1430, 4862]
>>> drevesa(100)
896519947090131496687170070074100632420837521538745909320
```

5. naloga (25 točk)