

Računalništvo (fizika): 2. izpit

9. junij 2020

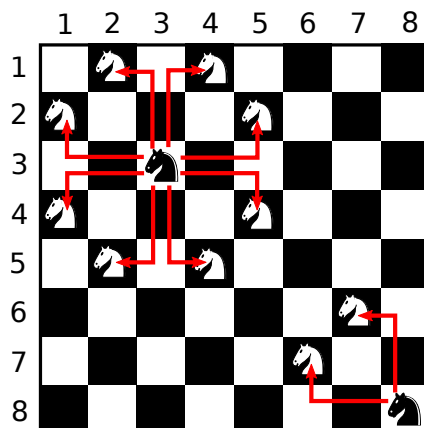
Naloge rešujte na strežniku Tomo, <https://www.projekt-tomo.si>.

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)

Polja na šahovnici velikosti 8×8 označimo s koordinatami (x, y) , kjer je $1 \leq x \leq 8$ in $1 \leq y \leq 8$, kakor je prikazano na sliki, ki ponazarja tudi premike šahovskega skakača.



Sprehod šahovskega skakača je tako zaporedje *veljavnih* koordinat na šahovnici $[(x_1, y_1), \dots, (x_n, y_n)]$, da lahko skakač skoči s polja (x_i, y_i) na (x_{i+1}, y_{i+1}) , za vse $1 \leq i < n$. Sprehod sme vsebovati eno polje večkrat.

Sestavite funkcijo `sprehod(lst)`, ki vrne `True`, če je dano zaporedje parov števil `lst` sprehod šahovskega skakača, sicer vrne `False`. Primeri uporabe:

```
>>> sprehod([])
True
>>> sprehod([(1,2)])
True
>>> sprehod([(9,2)])
False
>>> sprehod([(1,2), (3,4), (4,6)])
True
```

2. naloga (25 točk)

Odvedljivo funkcijo f v Pythonu predstavimo s parom (f, f') , kjer je f' odvod funkcije f . Na primer, funkcijo $x \mapsto 3 * x^4 + 1$ predstavimo s parom funkcij

```
((lambda x: 3 * x ** 4 + 1), (lambda x : 12 * x ** 3))
```

funkcijo $x \mapsto \sin x$ pa s parom $(\text{math.sin}, \text{math.cos})$.

a) (10 točk) Sestavite funkcijo `poly(lst)`, ki sprejme seznam koeficientov $[a_0, a_1, \dots, a_n]$ in vrne urejni par (p, p') , kjer je $p(x) = a_0 + a_1x + \dots + a_nx^n$.

```
>>> p = poly([1.0, 0.0, 0.0, 0.0, 3.0])
>>> p
(<function <lambda> at 0x10f668650>, <function <lambda> at 0x10f6c6f50>)
>>> p[0](10)
30001.0
>>> p[1](10)
12000.0
```

b) (15 točk) Sestavite funkcijo `produkt(fs)`, ki sprejme seznam funkcij $fs = [(f_1, f'_1), \dots, (f_n, f'_n)]$ in vrne urejeni par (g, g') , pri čemer je $g(x) = f_1(x) \cdot f_2(x) \cdots f_n(x)$.

Na primer, takole bi izračunali produkt $x \mapsto (1+x)(1+2x)(1+3x^2)$ ter njegovo vrednost in odvod pri $x = 5$:

```
>>> g = produkt([poly([1.0, 1.0]), poly([1.0, 2.0]), poly([1.0, 0.0, 3.0])])
>>> g[0](5)
5016.0
>>> g[1](5)
3728.0
```

3. naloga (30 točk)

Podjetje Booty Bay Inc., opravlja javne dražbe. Na dražbah prodajajo predmete, za vsakega pa imajo na voljo ime, opis in izklicno ceno. V postopku dražbe morebitni kupci ponujajo ceno za predmet, a če kupec ponudi ceno, ki je nižja od izklicne ali ni višja od dosedaj najvišje, se njegova ponudba ne upošteva. Na koncu kupi predmet tisti, ki je podal najvišjo ceno.

a) (15 točk) Sestavite razred `Predmet`, ki predstavlja predmet na dražbi. Objekt naj ima attribute za ime, opis, izklicno ceno in ponudbe kupcev (ki jih na začetku še ni). Definirajte konstruktor, ki sprejme ime predmeta, njegov opis in izklicno ceno, ter ustrezni metodi `__str__` in `__repr__`.

b) (5 točk) Razredu `Predmet` dodajte metodo `cena`, ki vrne najvišjo ponujeno ceno, če je ta zabeležena, sicer pa izklicno ceno.

c) (5 točk) Razredu dodajte metodo `ponudba`, ki sprejme ime kupca in ponujeno ceno, ter ponudbo zabeleži. Ponudba se ne upošteva, če je manjša od izklicne cene ali če ni večja od do sedaj najvišje ponujene cene.

d) (5 točk) Razredu dodajte metodo `kupec`, ki vrne ime zmagovalnega kupca in njegovo ponujeno ceno. Če kupca ni, naj metoda vrne `None`.

4. naloga (30 točk)

Na vsako polje šahovnice velikosti $n \times m$ je princ Aljaž zapisal naravno število. Na polje $(1, 1)$, pri čemer uporabljamo oznake iz prve naloge, je postavil šahovskega skakača in rekel: "S skakačem moraš opraviti sprehod od polja $(1, 1)$ do polja (n, m) tako, da se na vsakem koraku nobena izmed koordinat ne zmanjša. V zaporu boš preživel toliko dni, kot bo seštevek števil na poljih, ki jih je obiskal skakač."

Kako bi potoval po šahovnici s skakačem, da boš čim manj dni preživel v zaporu?

a) (15 točk) Sestavite funkcijo `skakac(a)`, ki sprejme matriko `a`, ki vsebuje števila na šahovnici, in učinkovito izračuna najmanjše število jetniških dni. Če poti do polja (n, m) ni, naj funkcija vrne ∞ .

b) (15 točk) Dobrohotni kralj Andrej je skakača zamenjal s šahovsko damo, ki se premika navpično, vodoravno in diagonalno za poljubno število polj. Sestavite še funkcijo `dama(a)`, ki sprejme matriko `a`, ki vsebuje števila na poljih, in vrne najmanjše število jetniških dni. Tudi ta funkcija naj deluje učinkovito, v kodi pa pokomentirajte njeno časovno zahtevnost.