

# Računalništvo (MAFI): 1. kolokvij

5. marec 2008

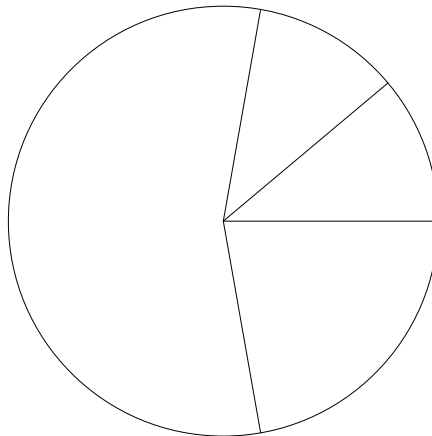
Čas reševanja: 120 minut

Reši vse štiri naloge. Pridobljenih 100 točk šteje za maksimalno oceno. Programe oddaj preko spletne učilnice, rešitve teoretičnih nalog pa na papirju.

## Naloga 1 [25 točk]

V Mathematici sestavi funkcijo `pita`, ki sprejme neprazno tabelo pozitivnih števil in vrne grafične ukaze za razrez kroga na izseke, ki prikazuje relativne deleže števil v tabeli glede na njihovo vsoto.

Na primer, ko poženemo `pita[{1, 1, 5, 2}]`, dobimo spodnjo sliko.



## Naloga 2 [25 točk]

Transformacija ravnine je *toga*, če ohranja razdalje in orientacijo. Znano je, da lahko vsako togo transformacijo zapišemo v obliki

$$\begin{pmatrix} x \\ y \end{pmatrix} \mapsto \begin{pmatrix} \cos \alpha & -\sin \alpha \\ \sin \alpha & \cos \alpha \end{pmatrix} \cdot \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}$$

Se pravi, toga transformacija je kompozitum rotacije za kot  $\alpha$  okoli izhodišča in translacije za vektor  $(b_1, b_2)^T$ . V Pythonu sestavi razred `Toga`, s katerim predstavimo togo transformacijo. Definiraj ustrezni konstruktor `__init__` in naslednje metode:

1. Metoda `kot(self)`, ki vrne kot rotacije  $\alpha$ .
2. Metoda `translacija(self)`, ki vrne vektor translacije  $(b_1, b_2)$  kot urejeni par.
3. Metoda `__repr__` ki predstavi transformacijo kot niz. Zaželeno je, da je niz oblike `"(α, b1, b2)"`, se pravi, da ne prikažemo celotne matrike, ampak samo kot  $\alpha$ .
4. Metoda `kompozicija(self, t)`, ki izračuna kompozicijo transformacij `self` in `t`.
5. **Dodatna naloga [+10 točk]:** dodaj še ustrežno metodo, da se bo objekt razreda `Toga` obnašal kot funkcija v Pythonu. Če je na primer `t` objekt razreda `Toga`, naj `t(x, y)` izračuna kam `t` preslika vektor  $(x, y)^T$ .

### Naloga 3 [10 + 15 točk]

Dana je tabela števil  $[a_1, \dots, a_n]$ . Pravimo, da je element  $a_i$  *lokalni minimum*, če velja  $a_{i-1} \leq a_i \leq a_{i+1}$  (robna elementa sta lokalna minimuma, če velja  $a_0 \leq a_1$  oziroma  $a_n \leq a_{n-1}$ ).

1. V Pythonu sestavi funkcijo `vsi_minimumi`, ki sprejme tako tabelo in vrne seznam vseh lokalnih minimumov. Oцени časovno zahtevnost funkcije v najslabšem primeru. Uporabi notacijo “veliki  $O$ ”.
2. V Pythonu sestavi funkcijo `minimum`, ki sprejme tako tabelo in vrne en (katerikoli) lokalni minimum. Oцени časovno zahtevnost funkcije v najslabšem primeru. Uporabi notacijo “veliki  $O$ ”.  
**Opomba:** za polno oceno mora funkcija delovati bistveno hitreje od funkcije `vsi_minimumi`, za dovolj velike tabele.

### Naloga 4 [15 + 10 točk]

Dano je zaporedje naravnih števil  $a = [a_1, \dots, a_n]$  dolžine  $n$  in naravno število  $r$ . Naša naloga je poiskati čim daljše podzaporedje (ne nujno strnjeno), katerega vsota je deljiva z  $r$ . Takih podzaporedij je lahko več, lahko pa je najdaljše tako zaporedje prazno.

1. Sestavi funkcijo `podzaporedje3(a)`, ki vrne najdaljše podzaporedje, katerega vsota je deljiva s 3. Oцени časovno zahtevnost funkcije v najslabšem primeru.
2. Sestavi funkcijo `podzaporedje(a, r)`, ki vrne najdaljše podzaporedje, katerega vsota je deljiva z  $r$ . Oцени časovno zahtevnost funkcije v najslabšem primeru.

**Dodatna naloga [+10 točk]:** Funkcija `podzaporedje(a, r)` naj deluje v  $O(r \cdot n)$  korakih.