

Računalništvo (fizika): 3. izpit

12. 6. 2017

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)

Zbirko besed lahko predstavimo z vgnezenimi slovarji takole:

- ključi slovarja so prve črke besed, ki nastopajo v zbirki
- vrednost pri danem ključu k je slovar, ki ga dobimo, če nadaljujemo z enakim postopkom na podzbirki vseh besed, ki se začnejo s k , pri čemer smo iz njih odstranili prvo črko.

Na primer, iz zbirke ['pet', 'pel', 'pot', 'tim', 'tia', 'teo'] dobimo slovar

```
{ 'p': { 'e': { 'l': {} },
        't': {} },
  'o': { 't': {} } },
{ 't': { 'i': { 'a': {} },
        'm': {} },
  'e': { 'o': {} } }
```

Sestavite *generator* zbirka(d), ki sprejme slovar d. Slovar predstavlja zbirko besed, kot je to opisano zgoraj. Generator naj vrača besede, ki so predstavljene z d.

2. naloga (30 točk)

Funkcija $f: \mathbb{N}_0 \rightarrow \mathbb{Z}$ je *strogo naraščajoča*, če velja $f(n) < f(n+1)$ za vse $n \in \mathbb{N}_0$.

a) (10 točk) Sestavite funkcijo nicla(f), ki sprejme strogo naraščajočo funkcijo f in poišče njeno ničlo, se pravi, da vrne tisto število $n \in \mathbb{N}_0$, za katerega je $f(n) = 0$. Če tako število ne obstaja, naj funkcija vrne None.

b) (20 točk) Sestavite še funkcijo hitra_nicla, ki deluje enako kot nicla, vendar s časovno zahtevnostjo $O(\log n)$, kjer je n ničla dane funkcije. V primeru, da ničla ne obstaja, časovna zahtevnost ni pomembna (lahko pa v komentarju navedete, kakšna je v tem primeru).

3. naloga (55 točk)

Fizik je izvedel meritve in dobil zaporedje realnih števil $x_0, x_2, \dots, x_{n-1}$. Poiskati želi taka indeksa $0 \leq i \leq j \leq n$, da je vsota podzaporedja $x_i + x_{i+1} + \dots + x_{j-1}$ čim manjša. (Lahko se zgodi, da je $i = j$, podzaporedje prazno in vsota 0, če so vsi elementi zaporedja pozitivni.) Pomagali mu bomo implementirati algoritem, ki deluje v času $O(n \log n)$.

a) (10 točk) Najprej obravnavajmo primer, ko je vrednost j v naprej dana. Sestavite funkcijo podzaporedjeD(x,j), ki sprejme seznam $x = [x_0, \dots, x_{n-1}]$ in število $0 \leq j \leq n$ ter vrne tisti $0 \leq i \leq j$, pri katerem zavzame vsota $x_i + x_{i+1} + \dots + x_{j-1}$ minimalno vrednost. Funkcija naj deluje v času $O(n)$.

b) (15 točk) Sestavite funkcijo podzaporedjeSlabo(x), ki sprejme seznam x in vrne tak par (i, j) , da je vsota $x_i + x_{i+1} + \dots + x_{j-1}$ minimalna. Funkcija naj deluje v času $O(n^2)$. Namig: uporabite funkcijo podzaporedjeD.

c) (30 točk) Sestavite funkcijo podzaporedjeDobro(x), ki sprejme seznam x in vrne tak par (i, j) , da je vsota $x_i + x_{i+1} + \dots + x_{j-1}$ minimalna. Funkcija naj deluje v času $O(n \log n)$. Namig: definirajte še funkcijo podzaporedjeL(x,i), ki je simetrična funkciji podzaporedjeD, le da deluje za v naprej določeno vrednost i . Nalogo rešite s postopkom deli in vladaj.