

Računalništvo (Fizika): 2. izpit

10. september 2013

Čas reševanja: 120 minut. Doseženih 100 točk šteje za maksimalno oceno.

Rešitev je potrebno oddati na spletni učilnici kot eno ZIP ali 7z datoteko, ki vsebuje rešitve posameznih nalog v ločenih datotekah. Posamezne datoteke poimenuj `PrimekIme_Nx`, kjer je x zaporedna številka naloge.

Naloga 1 [25+5 točk]

Pripravi delovni zvezek v *Mathematici*, v katerem rešiš naslednjo nalogo:

Naj bo $f(x) = \sqrt{\frac{x^3}{x-1}}$.

- (a) Nariši graf funkcije f na tistem delu definicijskega območja, ki le znotraj intervala $[-5, 5]$.
- (b) Z grafa lahko razbereš, da ima funkcija nekaj asimptot. Z *Mathematico* določi enačbe vseh poševnih asimptot.
- (c) Dodatna naloga [+5 točk]: Graf funkcije opremi z asimptotami.

Naloga 2 [25+5 točk]

Dan je seznam besed. Pravimo, da sta besedi a in b *sosednji*, če se zadnja črka besede a ujema s prvo črko besede b .

- (a) V Pythonu sestavi funkcijo `staSosednji(seznam, a, b)`, ki ugotovi, ali obstaja tako zaporedje sosednjih besed iz danega seznama, ki se začne z a , konča pa z b . Uporabi strategijo sestopanja.
- (b) Dodatna naloga [+5 točk]: Sestavi funkcijo `najkrajšaPot(seznam, a, b)`, ki vrne najkrajše zaporedje sosednjih besed iz danega seznama, ki se začne z a , konča pa z b . Če tako zaporedje ne obstaja, naj funkcija vrne `None`.

(obrni list)

Naloga 3 [25+5 točk]

Dvojiško drevo je podano z razredom `Drevo`. V vsakem vozlišču drevesa je zapisano celo število ali pa znak `V` ali znak `Z`. Drevo tako na naraven način prikazuje izračun nadomestne upornosti vezja uporov, kjer so upori lahko vezani vzporedno ali zaporedno.

- (a) Napiši konstruktor `__init__(self, a, levo, desno)`, kjer je `a` celo število ali `V`, `Z`, medtem ko sta `levo` in `desno` lahko `None` ali `Drevo`. Če je kombinacija vhodnih parametrov nesmiselna, vrži izjemo!
- (b) Dodaj metodo `izracunaj(self)`, ki vrne vrednost računa, ki ga drevo predstavlja.
- (c) Napiši metodo `__repr__(self)`, ki izpiše račun, ki ga drevo predstavlja.
- (d) Dodatna naloga [+5 točk]: Dodaj metodo `poisciVezje(self, r, eps)`, ki v drevesu poišče (kakšno) poddrevo, katerega vezje, ki ga poddrevo predstavlja, ima upornost, ki se od vrednosti `r` razlikuje največ za `eps`.

Naloga 4 [25 točk]

Dana je velika tabela n celih števil.

- (a) Napiši metodo `imamoOdgovor(tabela)`, ki za dano tabelo na čimbolj učinkovit način ugotovi, ali obstaja trojica a_i, a_j, a_k elementov tabele z različnimi indeksi, da je njihova vsota enaka 42. Utemelji časovno zahtevnost tvoje rešitve.
- (b) Napiši metodo `imamoOdgovor2(tabela)`, ki reši zgoraj naveden problem v časovni zahtevnosti strogo manjši od $O(n^3)$. Utemelji časovno zahtevnost.