Računska zahtevnost

prva domača naloga

Rok za oddajo domače naloge je torek, 12. 11. 2013 ob 16.00. Oddaja je preko spletne učilnice ali v predalček asistenta (pritličje na Jadranski 19 (fizika)). Če imate vprašanja, se obrnite na asistenta ali profesorja oz. uporabite forum na učilnici. O nalogah se lahko pogovarjate, o rešitvah pa ne. Če boste uporabili vire (knjige, splet), jih prosimo navedite.

Naloga 1 (4 točke)

Za $x = x_{n-1}x_{n-2}\dots x_0 \in \{0,1\}^n$ naj bo $x_{(2)} = \sum_{i=0}^{n-1} x_i 2^i$ število, ki ga beseda x predstavlja v dvojiškem zapisu, npr. $1101_{(2)} = 13$. Definirajmo jezik $J = \{x2y;\ x, y \in \{0,1\}^*, x, y \text{ se začneta z } 1, x_{(2)} = y_{(2)} + 2\}$. Velja npr. $1121 \in J$, $10012111 \in J$, $1020 \notin J$. Sestavite Turingov stroj, ki odloči, ali dana beseda $w \in \{0,1,2\}^*$ pripada jeziku J!

Kakšna je časovna zahtevnost (v notaciji veliki O) vašega Turingovega stroja?

Naloga 2 (2 točki)

Naj bosta f, g funkciji iz $\mathbb{N}$ v $\mathbb{R} \cap (0, \infty)$. Ali naslednji trditvi držita (odgovor utemelji):

- $\lim\limits_{n\to\infty} \dfrac{g(n)}{f(n)} = 0 \implies f(n) \neq O(g(n)),$
- $f(n) = \Theta(g(n)) \implies \lim\limits_{n\to\infty} \dfrac{g(n)}{f(n)} = O(1).$

Naloga 3 (4 točke)

Opišite algoritem linearne časovne zahtevnosti (t.j. $O(n)$), ki na vhodu $w \in \mathbb{N}$ vrne tak $x \in \mathbb{N}$, da je $5^x = w$. Če w ni potenca petice, lahko algoritem vrne poljubno število. Števila so kodirana v dvojiškem sistemu (vhod in izhod sta nad abecedo $\Sigma = \{0,1\}$).

Znotraj algoritma lahko uporabljate operacije, za katere smo na predavanjih ali vajah povedali, kako so zahtevne (npr. računanje determinante, seštevanje, množenje) in uporabljate izpeljane ocene za število korakov (npr.

za množenje $O(n^2)$). Dodatno lahko uporabite dejstvo[1], da gre $\log_2 5$ na t mest natančno izračunati v času, ki je polinomski v t. Algoritem naj bo tako natančno opisan, da bo moč utemeljiti linearno časovno zahtevnost. To tudi storite.

Namig 1: Če w ni potenca petice, lahko algoritem vrne poljubno število. Torej je dovolj, da dovolj dobro ocenimo x.

Namig 2: Algoritem časovne zahtevnosti $O(n \log n)$ je vreden 2 točki.

[1] Glej http://en.wikipedia.org/wiki/Computational_complexity_of_mathematical_operations