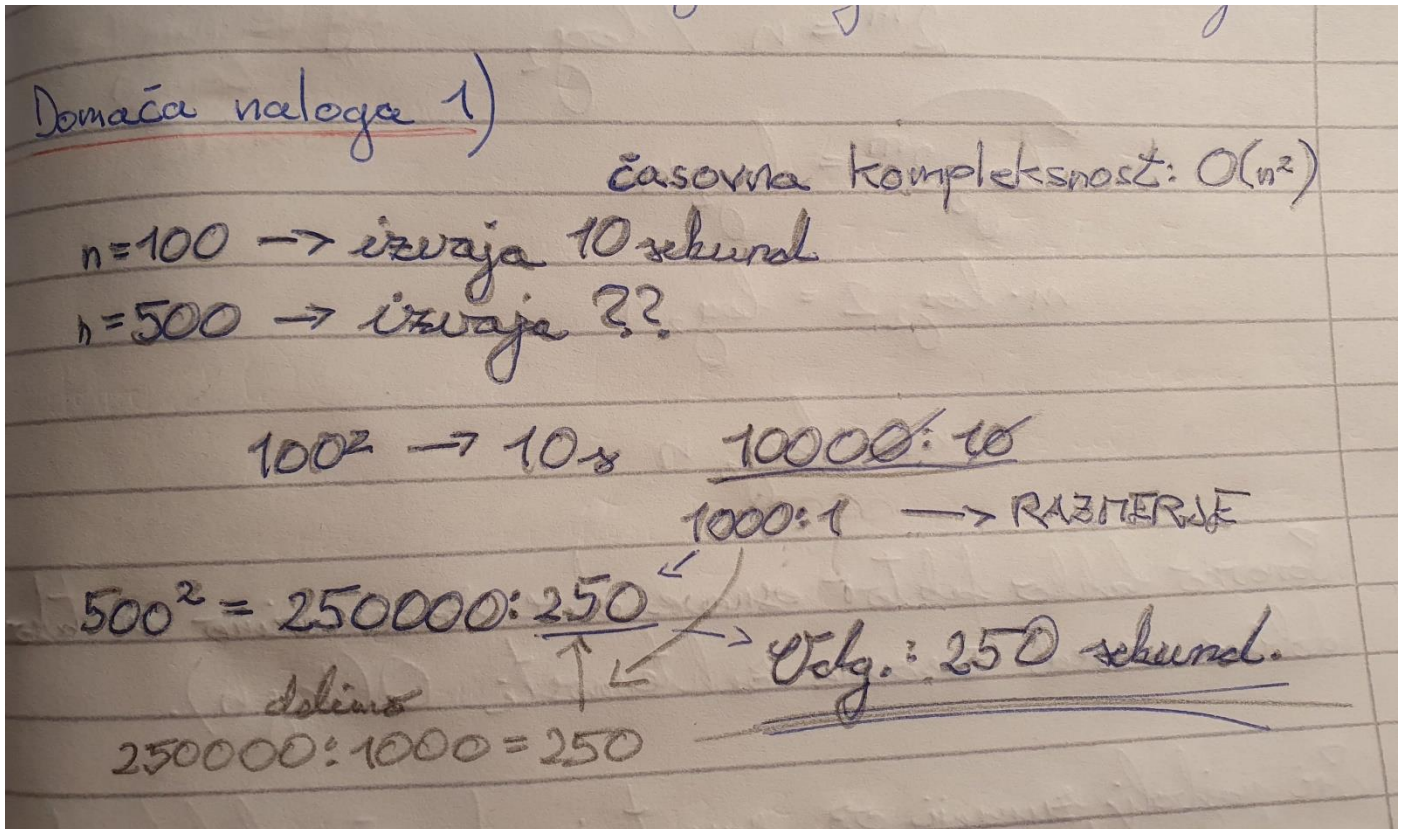


VAJE 9 – Domača naloga

Domača naloga 1

Algoritem, katerega časovna zahtevnost je reda velikosti $\Theta(n^2)$, se pri vrednosti $n = 100$ na določenem računalniku izvaja 10 sekund. Kako dolgo naj bi se izvajal pri vrednosti $n = 500$?



Domača naloga 3

V kateri razred zahtevnosti sodi algoritem?

Utemelji odgovore!

Preberi vrednost n

$i = n$

Dokler $i > 0$, ponavlajaj

$i = i / 2$

$j = 1$

Dokler $j < n$, ponavlajaj

$j = j * 2$

$k = 0$

Dokler $k < n$, ponavlajaj

$k = k + 4$

(konstantno število operacij)

Ustavi se.

Domača naloga 3)

Algoritem sedi v red kompleksnosti $O(n * (\log(n))^2)$.

Utemeljitev:

Notraj algoritma imamo tri ränge. Isto notranja zmanjšanje k privedi vrednost $k+1$, kar pomeni, da je število operacij enako $n/4$. Faktor $1/4$ lahko poiščemo ugotavljanje časovne kompleksnosti računamo, tako da je razred časovne zahtevnosti te ränge $O(n)$.

Upam, da boste nalogo brez problema prebrali, v nasprotnem primeru pa bom besedilo prepis še na računalnik.

Rad bi samo za vsak slučaj opozoril, da je kot rešitev mišljena $O(n * (\log(n))^2)$ torej na kvadrat je postavljen celoten $\log(n)$ in ne samo n . Torej v bistvu tole:

$$\log^2(n)$$

Somakrmo se ränge višje, kjer imamo priveditveni stavček $j = j * 2$. V tem primeru se izvede 2^m število operacij (m je število po katerem je $2^m \geq n$). Ker mi želimo število ~~po~~ korakov izraziti z n , enačimo $2^m \geq n$, kar se nato znebimo eksponentov. Torej obe strani logaritmiramo z logaritmom osnove 2:

$$2^m = n \quad / \log_2$$

saj se nam bo
potem 2
okrajšala

$$\log_2 2^m = \log_2 n$$

$$m \cdot \log_2 2 = \log_2 n$$

$$m = \log_2 n$$

Podoben lahko faktor osnove dve računavamo in tako dobimo razred časovne zahtevnosti $O(\log n)$.

Pri najbolj zunanji oz. prvotni rangu se obdelavnik ränge je situacija v bistvu enaka, le da imamo namesto množitve deljenje z 2 - ta ranga prav tako sedi v logaritmični razred. Sedaj lahko te med seboj zmanjšamo ter tako dobimo $n * \log(n) * \log(n)$ oz. $n * \log(n)^2$. Torej da je rešitev $O(n * (\log(n))^2)$.

Domača naloga 2

V kateri razred zahtevnosti sodi algoritem v primeru, da:

- a) Za osnovno enoto dela izberemo odštevanje?
- b) Za osnovno enoto dela izberemo celoštevilsko deljenje?
- c) Za osnovno enoto dela izberemo seštevanje?

Utemelji odgovore!

Preberi vrednost n

Nastavi c na 0

Nastavi a na vrednost n

Dokler $a \geq 1$, ponavljaj

$c = c + 1$

$b = n$

Dokler $b > 0$, ponavljaj

$c = c + 1$

$b = b / 2$

$a = a - 1$

Ustavi se.

Naloga 2.)

a) V primeru, da za osnovno enoto dela izberemo odštevanje, sodi algoritem v razred zahtevnosti $O(n)$. Gledamo namreč samo zunanjo zanko, saj se samo v tej nahaja odštevanje. V tej zanki imamo 3 preizvedene stavke, ki se izvedejo v n -korakih, torej dobimo $3n$. Faktor 3 lahko izpustimo, ter tako dobimo razred $O(n)$.

b) V tem primeru sodi algoritem v razred časovne zahtevnosti $O(n * \log(n))$. Operacija celoštevilskega deljenja se nahaja v notranji zanki, zato upoštevamo to ter vse njene "parent" zanke. V našem primeru tako upoštevamo celoten algoritem. Za zunanjo zanko smo že ugotovili da pada v $O(n)$, notranja pa zaradi deljenja z 2 v $O(\log(n))$ - to sem podrobneje razložil pri 3. nalogi. Če preizvedene stavke za vsako notranjo zanko se nam pojavijo še v zunanji zanki, zato ga lahko obravnavamo kot faktor ter zanemarimo. Nato samo n in $\log(n)$ zmnožimo med seboj.

c) Kakor v b) poimava tudi ~~da~~ pri osnovni enoti
seštevanja upoštevamo celoten algoritem, torej spada
tudi ta v red $O(n * \log(n))$.

- Tilen Padovnik, vpisna št.: 63190211