ALGORITMI I STRUKTURE PODATAKA

I čas

1 Analiza algoritama

1.1 Oznake O, Ω, Θ, o

Neka su f i g pozitivne funkcije od argumenta n iz skupa N. Kažemo da je $\boldsymbol{g(n) = O(f(n))}$ ako postoje pozitivne konstante c i N tako da $\forall\ n \geq N$ važi: $g(n) \leq c \cdot f(n)$.

Uočimo i da O(1) je oznaka za klasu ograničenih funkcija.

Takođe važi:
$O(f(n)) + O(g(n)) = O(f(n) + g(n))$
$O(f(n)) \cdot O(g(n)) = O(f(n) \cdot g(n))$

Oznaka O(f(n)) se u stvari odnosi na klasu funkcija, a jednakost g(n)=O(f(n)) je oznaka za inkluziju g(n)∈O(f(n)).

Primer: $5n^2 + 8 = O(n^2)$
Jer je $5n^2 + 8 <= 6n^2$ za $n>=3$

Ovim dobijamo da možemo zanemarivati multiplikativne konstante, tj**. O(3n+2) pišemo kao O(n)**

Dalje, osnova logaritma nije bitna, jer je $\log_a n = \log_a (b^{\log_b n}) = \log_b n \underbrace{\log_a b}_{const}$

Kažemo da je funkcija $f(n)$ asimptotska donja granica funkcije $g(n)$ i pišemo $\boldsymbol{g(n) = \Omega(f(n))}$ ako postoje pozitivne konstante c i N tako da $\forall\ n \geq N$ važi: $g(n) > c \cdot f(n)$.

Ako za funkcije $f(n)$ i $g(n)$ istovremeno važi: $f(n) = O(g(n))$ i $f(n) = \Omega(g(n))$ onda pišemo $\boldsymbol{f(n) = \Theta(g(n))}$.

1. Dokazati da važi: **$T(n)=2n^2+n-1 \quad = \quad \Theta(n^2)$**
DOKAZ: Potrebno je dokazati da postoje **c_1, c_2,N,** $\forall\ n \geq N$ važi: $c_1 n^2 <= 2n^2+n-1 <= c_2 n^2$ **,** to jest

$$c_1 <= 2 + \frac{1}{n} - \frac{1}{n^2} <= c_2$$

Poslednja nejednakost važi za $c_1=2$, $c_2=3$, N=1

Slično, važi i da:
$T(n)=2n+3 \quad = \quad \Theta(n)$
$T(n)=10+3\log n \quad = \quad \Theta(\log n)$
$T(n)=2^n + n^3 - 100 = \Theta(2^n)$

2. Dokazati da važi: $17n \log n - 23n - 10 = \Theta(n \log n)$
DOKAZ: Potrebno je dokazati da postoje **c_1, c_2** tako da: $c_1 n \log n <= 17n \log n - 23n - 10 <= c_2 n \log n$ **,** to jest deljenjem sa n log n

$$c_1 <= 17 - \frac{23}{\log n} - \frac{10}{n \log n} <= c_2$$

Poslednja nejednakost važi za $c_1=4$, $c_2=17$, N=4 (proverite neposredno)

Ako za funkcije $f(n)$ i $g(n)$ važi: $\lim_{n \to \infty} \frac{f(n)}{g(n)} = 0$, pišemo: $f(n) = o(g(n))$.

3. Odrediti vreme izvršavanja sledećeg programskog fragmenta. Rešenje prikazati polinomijalnim izrazom i u O notaciji.

```
i=0;  k=1;
while (k<=n)
{
   k=2*k;
   i=i+1;
}
```

Rešenje: T(n)=1+1+log n + 2*log n + 2*log n =2 + 5*log n = O(log n)
Zašto? Odredite broj izvršenih operacija u svakoj liniji prethodnog programskog fragmenta.

Broj operacija u liniji 1 je 2. Zašto?
Broj operacija u liniji 2 je log n. Zašto? Koliko puta se ponavlja while ciklus i poređenje k<=n?
Pomoć: Uslov za izlazak iz ciklusa je k>n.
Kako se menja promenljiva k? Na početku k=1.
Nakon prvog ponavljanja cilkusa $k=2*1=2=2^1$
Nakon drugog ponavljanja cilkusa $k=2*2=4=2^2$
Nakon trećeg ponavljanja cilkusa $k=4*2=8=2^3$
. . .
Nakon m-tog ponavljanja cilkusa $k=2*2^{m-1}=2^m$
Uslov za izlazak iz ciklusa je $k=2^m>n$, odnosno nakon m-tog koraka kada se dobije vrednost $2^m>n$,
tj. kada postane $m > \log_2 n$

Koliki je broj operacija u linijama 4,5?

4. **(Za vežbu)** Odrediti vreme izvršavanja sledećeg programskog fragmenta. Rešenje prikazati polinomijalnim izrazom i u O notaciji.

```
for (i=1, j=0; i<=n; i<<1) j++;
```

(Za vežbu) Princip matematičke indukcije
5. Dokazati:

(a) $\sum_{i=1}^{n} i = \frac{n(n+1)}{2}$

(b) $\sum_{i=0}^{n} q^i = \frac{q^{n+1}-1}{q-1}$

(c) $\sum_{i=1}^{n} i \cdot 2^i = (n-1)2^{n+1} + 2$

(d) $\sum_{i=1}^{n} i \cdot 2^{n-i} = 2^{n+1} - n - 2$

(e) Dokazati da važi: $\lfloor \log n \rfloor = O(n)$
Podsećanje: Neka $x \in \mathbb{R}$. Važi da: $x-1 < \lfloor x \rfloor <= x <= \lceil x \rceil < x+1$
Koliko je $\lfloor 8 \rfloor$? Koliko je $\lceil 8 \rceil$? Koliko je $\lfloor 7.51 \rfloor$? Koliko je $\lceil 7.51 \rceil$?
Neka $n \in \mathbb{N}$. Koliko je $\lfloor n/2 \rfloor + \lceil n/2 \rceil$?

1.2 Rekurentne relacije

6. Rešiti diferencnu jednačinu: $T(n) = 5T(n-1) - 8T(n-2) + 4T(n-3),\ n \geq 4$
za koju važi $T(1) = 2,\ T(2) = 4,\ T(3) = 12$.

Rešenje: $T(n)=4 + (-2+n)2^n$

7. Rešiti rekurentnu jednačinu: $T(n) = 3T(n-1) + 2,\ n \geq 2$ za koju važi $T(1) = 1$.
Rešenje: $T(n) = 2*3^{n-1}-1$ Zašto?

Naime,
$T(n) = 3T(n-1) + 2$
$T(n\text{-}1) = 3T(n-2) + 2$
=================
Oduzimanjem sledi: T(n)-T(n-1)=3T(n-1)-3T(n-2)
Tj. dobija se homogena j-na: T(n) -4T(n-1)+3T(n-2)=0, $T(1) = 1$

Jednačine zasnovane na dekompoziciji

U slučaju kada se obrada ulaza veličine **n** svodi na **a** obrada ulaza veličine $(\frac{n}{b})$ i potrebno je izvršiti još $c \cdot n^k$ koraka, jednačina izgleda ovako: $T(n) = aT(\frac{n}{b}) + c \cdot n^k$ gde *a, b, c, k >=0, b !=0* i dato je T(1).

Master teorema: Rešenje diferencne jednačine: $T(n) = aT(\frac{n}{b}) + c \cdot n^k$ gde *a, b, c, k >=0, b !=0* je:

$$T(n) = \begin{cases} O(n^{\log_b a}), a > b^k \\ O(n^k \log n), a = b^k \\ O(n^k), a < b^k \end{cases}$$

Primer: T(1)=1, T(n)= 2 T(n/2) + O(n)

Kako je a=2, b=2, k=1, onda je $a=b^k$, te je po master teoremi: T(n) = O (n log n)

8. Odrediti asimptotsko ponašanje rešenja $T(n)$ diferencne jednačine:

$$T(n) = T(\frac{n}{2}) + \sqrt{n}, \quad n \geq 2$$

za koju važi $T(1) = 1$. Može se smatrati da n uzima samo vrednosti jednake stepenima dvojke.

Rešenje: a=1, b=2, k=$\frac{1}{2}$ => $a<b^k$ => po master teoremi $T(n)=O(n^k)=O(\sqrt{n})$

9. Rešiti rekurentnu jednačnu:
$T(n) = 2T(n/2) + c \log_2 n, \ n \geq 2$
za koju važi $T(1) = 0$. Moˇze se pretpostaviti da je n oblika 2^k.

10. Neka je data rekurentna jednačina za vreme izvršavanja T(n) algoritma BINARNA_PRETRAGA:
T(1)=1,

$T(n)=T(\lfloor n/2 \rfloor) + 1, n>1$

Dokažite indukcijom po n da je za svako n>=1 ispunjeno $T(n)=1+\lfloor \log n \rfloor$

Rešenje:

Baza: osnovni slučaj n=1 je zadovoljen, jer je $T(1)=1+\lfloor \log 1 \rfloor=1+0=1$

Induktivna hipoteza (IH): Pretpostavimo da je rekurentna jednačina tačna za $\lfloor n/2 \rfloor$.
Dokazimo da je tačna za n>1.

Dakle, po IH važi da $T(\lfloor n/2 \rfloor)=1+\lfloor \log \lfloor n/2 \rfloor \rfloor$

Onda je: $T(n)= T(\lfloor n/2 \rfloor) + 1 =$

(po IH) $= 1+\lfloor \log \lfloor n/2 \rfloor \rfloor +1$

$= 1+\lfloor \log \lfloor n/2 \rfloor +1 \rfloor$

$= 1+\lfloor \log \lfloor n/2 \rfloor +\log 2 \rfloor$

$= 1+\lfloor \log 2*\lfloor n/2 \rfloor \rfloor$ (***)

Ako je n paran broj, onda važi da je $2*\lfloor n/2 \rfloor=\lfloor 2*n/2 \rfloor$, te se neposrednim izvođenjem u (***) dobija da je tada $T(n)=1+\lfloor \log n \rfloor$

Ako je n neparan broj, onda važi da je $2*\lfloor n/2 \rfloor=\lfloor 2*n/2 \rfloor-1$, te se neposrednim izvođenjem u (***) dobija da je tada $T(n)=1+\lfloor \log (n-1) \rfloor$. Kako važi da $\lfloor \log (n) \rfloor \neq \lfloor \log (n-1) \rfloor$ samo kada je broj n jednak stepenu broja 2, onda za neparno n važi da $\lfloor \log (n) \rfloor = \lfloor \log (n-1) \rfloor$.

Odatle sledi da za neparno n važi: $T(n)=1+\lfloor \log (n-1) \rfloor=1+\lfloor \log (n) \rfloor$ QED