Teorija izračunljivosti

Ivan Prokić
kabinet 117, F blok
prokic@uns.ac.rs
http://imft.ftn.uns.ac.rs/∼iprokic/

Novi Sad

Tema 1

Rast funkcija, asimptotske notacije

Asimptotska efikasnost algoritama

Šta radimo?

- Kao što smo rekli, nećemo računati precizno vreme rada algoritma, već samo davati ocenu za najgore moguće vreme rada, i to asimptotski.
- Razlog je što za dovoljno veliki ulaz konstante i termi sporijeg rasta budu ”dominirani” od strane veličine inputa.

Zašto to radimo?

- Da bi ocenili efikasnost samog algoritma.
- Da bi mogli porediti efikasnost različitih algoritama (koji rešavaju isti problem).

Asimptotske notacije

Funkciju vremena rada algoritma posmatramo kao $f : \mathbb{N} \to \mathbb{N}$, tj. veličina ulaza je prirodan broj i vreme rada algoritma je takođe prirodan broj.

Asimptotske notacije:

- Θ-notacija (tesna granica);
- O-notacija (gornja granica);
- Ω-notacija (donja granica);
- o-notacija (gornja granica koja nije asimptotski tesna);
- ω-notacija (donja granica koja nije asimptotski tesna).

Θ-notacija (tesna granica)

Definicija (Θ-notacija)

Za datu funkciju $f(n)$, skup funkcija $\Theta(f(n))$ je definisan sa

$$\begin{aligned}\Theta(f(n)) = \quad & \{g(n) \,:\, (\exists c_1, c_2 \in \mathbb{R}^+)(\exists n_0 \in \mathbb{N})(\forall n \geq n_0) \\ & 0 \leq c_1 f(n) \leq g(n) \leq c_2 f(n)\}.\end{aligned}$$

Zadatak (Termi nižeg reda "nisu važni")

Pokazati da $2n^2 + 1 \in \Theta(n^2)$.

Rešenje.

$$c_1 n^2 \leq 2n^2 + 1 \leq c_2 n^2 \quad \Leftrightarrow \quad c_1 \leq 2 + \frac{1}{n^2} \leq c_2$$

Možemo uzeti $n_0 = 1, c_1 = 2$ i $c_2 = 3$. $\diamond$

Θ-notacija: primeri

Zadatak

Pokazati da $an^2 + bn + c \in \Theta(n^2)$*, gde su* $a, b, c \in \mathbb{R}$ *i* $a > 0$*.*

Rešenje.

$$c_1 n^2 \leq an^2 + bn + c \leq c_2 n^2 \quad \Leftrightarrow \quad c_1 \leq a + \frac{b}{n} + \frac{c}{n^2} \leq c_2 \qquad (1)$$

c_1 i c_2 ćemo izraziti preko a, jer je $a > 0$. Ako izaberemo $n_0 = 2 \cdot \max\left(\frac{|b|}{a}, \sqrt{\frac{|c|}{a}}\right)$, dobijamo da za sve $n \geq n_0$ važi $\frac{|b|}{n} \leq \frac{a}{2}$ i $\frac{|c|}{n^2} \leq \frac{a}{4}$. Odatle za navedeno n_0, birajući $c_1 = \frac{a}{4}$ i $c_2 = \frac{7a}{4}$ uslov dat u (1) važi za sve $n \geq n_0$. $\diamond$

Θ-notacija: primeri

Zadatak

Pokazati da $n \notin \Theta(n^2)$ *i* $n^3 \notin \Theta(n^2)$.

Θ-notacija: primeri

Zadatak

Pokazati da $n \notin \Theta(n^2)$ *i* $n^3 \notin \Theta(n^2)$.

Rešenje. Pretpostavimo da $n \in \Theta(n^2)$, tj. da postoje konstante $c_1, c_2 \in \mathbb{R}^+$ i $n_0 \in \mathbb{N}$ takvi da za sve $n \geq n_0$ važi

$$c_1 n^2 \leq n \leq c_2 n^2 \tag{2}$$

Iz poslednjeg sledi $c_1 \leq \frac{1}{n}$ za sve $n \geq n_0$, što je u kontradikciji sa činjenicom da je $c_1 > 0$ konstanta i $\lim\limits_{n \to \infty} \frac{1}{n} = 0$.

Pretpostavimo da $n^3 \in \Theta(n^2)$, tj. $c_1 n^2 \leq n^3 \leq c_2 n^2$. Odavde se dobija $n \leq c_2$, za sve $n \geq n_0$, što je u kontradikciji sa činjenicom da je c_2 konstanta.

$\diamond$

O-notacija (gornja granica)

Definicija (O-notacija)

Za datu funkciju $f(n)$, skup funkcija $O(f(n))$ je definisan sa

$$O(f(n)) = \{g(n) : (\exists c \in \mathbb{R}^+)(\exists n_0 \in \mathbb{N})(\forall n \geq n_0) \\ 0 \leq g(n) \leq cf(n)\}.$$

Zadatak (Termi nižeg reda "nisu važni")

Pokazati da $2n^2 + 1 \in O(n^2)$.

Rešenje. Sledi direktno iz $2n^2 + 1 \in \Theta(n^2)$. $\diamond$

O-notacija: primeri

Lema (Θ je strožija od O)

Iz $f(n) \in \Theta(g(n))$ *sledi* $f(n) \in O(g(n))$, *odnosno* $\Theta(g(n)) \subseteq O(g(n))$.

Zadatak

Pokazati da $n \in O(n^2)$ *i* $n^3 \notin O(n^2)$.

Rešenje. Pokažimo $n \in O(n^2)$. Kako je

$$n \leq cn^2 \quad \Leftrightarrow \quad \frac{1}{n} \leq c$$

Možemo uzeti $n_0 = 1$ i $c = 1$.

Pretpostavimo da $n^3 \in O(n^2)$, tj. $n^3 \leq cn^2$. Odavde se dobija $n \leq c$, za sve $n \geq n_0$, što je kontradikcija.

$\diamond$

Ω-notacija (donja granica)

Definicija (Ω-notacija)

Za datu funkciju $f(n)$, skup funkcija $\Omega(f(n))$ je definisan sa

$$\begin{aligned}\Omega(f(n)) = \quad & \{g(n) \,:\, (\exists c \in \mathbb{R}^+)(\exists n_0 \in \mathbb{N})(\forall n \geq n_0) \\ & 0 \leq cf(n) \leq g(n)\}.\end{aligned}$$

Zadatak (Termi nižeg reda "nisu važni")

Pokazati da $2n^2 + 1 \in \Omega(n^2)$.

Rešenje. Sledi direktno iz $2n^2 + 1 \in \Theta(n^2)$. $\diamond$

Ω-notacija: primeri

Lema (Θ je strožija od Ω)

Iz $f(n) \in \Theta(g(n))$ *sledi* $f(n) \in \Omega(g(n))$*, odnosno* $\Theta(g(n)) \subseteq \Omega(g(n))$*.*

Zadatak

Pokazati da $n \notin \Omega(n^2)$ *i* $n^3 \in \Omega(n^2)$*.*

Rešenje. Analogno rešenju zadatka kod O-notacije (domaći). $\diamond$

Asimptotske notacije: osobine

Teorema

Za svake dve funkcije $f(n)$ *i* $g(n)$ *važi* $f(n) \in \Theta(g(n))$ *ako i samo ako* $f(n) \in O(g(n))$ *i* $f(n) \in \Omega(g(n))$.

Dokaz

Na času.

Teorema (Refleksivnost)

- $f(n) \in \Theta(f(n))$;
- $f(n) \in O(f(n))$;
- $f(n) \in \Omega(f(n))$;

Dokaz

Na času.

Asimptotske notacije: osobine

Teorema (Simetričnost)

$f(n) \in \Theta(g(n))$ *ako i samo ako* $g(n) \in \Theta(f(n))$.

Dokaz

Na času.

Teorema (Tranzitivnost)

- $f(n) \in \Theta(g(n)) \wedge g(n) \in \Theta(h(n)) \quad \Rightarrow \quad f(n) \in \Theta(h(n))$;
- $f(n) \in O(g(n)) \wedge g(n) \in O(h(n)) \quad \Rightarrow \quad f(n) \in O(h(n))$;
- $f(n) \in \Omega(g(n)) \wedge g(n) \in \Omega(h(n)) \quad \Rightarrow \quad f(n) \in \Omega(h(n))$;

Dokaz

Na času.

Asimptotske notacije: osobine

Teorema

$f(n) \in O(g(n))$ *ako i samo ako* $g(n) \in \Omega(f(n))$.

Dokaz

Na času.

Primer

Ako je $f(n) = n$ *i* $g(n) = n^{1+\cos n}$ *tada* $f(n) \notin O(g(n))$ *i* $g(n) \notin \Omega(f(n))$.

Teorema (Pravilo zbira)

$\max(f(n), g(n)) \in \Theta(f(n) + g(n))$.

Dokaz

Na času.

Asimptotske notacije: zadaci

Napomena: Umesto $f \in \Theta/O/\Omega$ pisaćemo i $f = \Theta/O/\Omega$.

Zadatak

Pokazati:

1. $0.0001 \cdot n^3 = \Theta(n^3)$;
2. $(an+b)^c = \Theta(n^c)$, *za* $a, b, c \in \mathbb{N}\backslash\{0\}$;
3. $\log_a n = \Theta(\log_b n)$, *za* $a, b > 0$ *i* $a, b \neq 1$;
4. $\log\log n = O(\log n)$;
5. $\log n = O(\sqrt{n})$;
6. $n \log n = O(n^2)$;
7. $n \log n \neq \Theta(n^2)$;
8. $3^n + n = \Theta(3^n)$;
9. $n! \neq O(2^n)$;

Rešenje. Na času. $\diamond$

Tema 2

Divide and conquer algoritmi

Merge sort

Podeliti početni problem da više potproblema, rešiti ih **rekurzivno** i na kraju **spojiti** rešenja potproblema u rešenje početnog problema.

Primer (Sortiranje niza)

Ulaz: *Niz A od* n *brojeva* $<a_1, a_2, \ldots, a_n>$
Izlaz: *Niz* $<a'_1, a'_2, \ldots, a'_n>$ *dobijen permutacijom elemenata ulaznog niza, tako da važi* $a'_1 \leq a'_2 \leq \ldots \leq a'_n$.

Merge sort:

- **Podeliti** niz od n elemenata na dva podniza od $\frac{n}{2}$ elemenata.
- **Rekurzivno** sortirati dva podniza.
- **Spojiti** dva sortirana podniza u traženi sortirani niz.

Merge sort - algoritam

Algorithm 1: Merge (A, p, q, r)

$n_1 = q - p + 1$
$n_2 = r - q$
$L[1 \dots n_1 + 1]$ i $R[1 \dots n_2 + 1]$ novi nizovi
  for $i = 1$ **to** n_1 **do**
    $L[i] = A[p + i - 1]$

for $j = 1$ **to** n_2 **do**
    $R[i] = A[q + j]$

$L[n_1 + 1] = \infty$
$R[n_2 + 1] = \infty$
$i = 1, j = 1$
for $k = p$ **to** r **do**
  if $L[i] \leq R[j]$ **then**
    $A[k] = L[i]$
    $i = i + 1$
  else
    $A[k] = R[j]$
    $j = j + 1$

Algorithm 2: Merge-Sort (A, p, r)

if $p < r$ **then**
  $q = \left\lfloor \frac{p+r}{2} \right\rfloor$
  Merge-Sort(A, p, q)
  Merge-Sort$(A, q + 1, r)$
  Merge(A, p, q, r)

- **Merge:** sortirane podnizove $A[p \dots q]$ i $A[q + 1 \dots r]$ **spaja** u jedan sortirani niz $A[p \dots r]$.
- **Merge-Sort**: sortira niz $A[p \dots r]$ tako što ga **podeli** na dva (skoro) jednaka podniza, koje zatim **rekurzivno** sortira, i na kraju **spoji** pomoću procedure Merge.
- Poziva se sa Merge-Sort$(A, 1, A.lenght)$.

Merge sort - analiza vremena rada

Primer

Skicirati način rada Merge-Sort algoritma na nizu $A = [1, 32, 43, 65, 11, 89, 32, 2]$.

Rešenje. Na času. $\diamond$

Vreme rada Merge-Sort: Za $n > 1$ (i n parno), vreme rada Merge-Sort je

$$T(n) = 2T(n/2) + D(n) + C(n),$$

gde je $D(n)$ vreme potrebno da se niz podeli, a $C(n)$ vreme potrebno da se podnizovi spoje. U našem slučaju $D(n) = \Theta(1)$, a $D(n)$ je vreme rada Merge(p, q, r) procedure, za $n = r - p + 1$. Prve dve petlje u Merge rade u vremenu $\Theta(n)$ (jer $n/2 - 1 < n_1, n_2 < n$), a treća radi u vremenu $\Theta(n)$, pa je $D(n) = \Theta(n)$. Dakle, vreme rada Merge-Sort algoritma je

$$T(n) = 2T(n/2) + \Theta(n).$$

Merge sort - primer

Napomena: Za bilo koje $n > 1$ vreme rada Merge-Sort algoritma je zapravo

$$T(n) = T(\lfloor n/2 \rfloor) + T(\lceil n/2 \rceil) + \Theta(n).$$

Primer

Za vreme rada Merge-Sort algoritma važi $T(n) = \Theta(n \lg n)$ *(na sledećem času), pa je Merge sort brži od Bubble sort (i Insertion sort i sl.) algoritma, koji je u* $\Theta(n^2)$*. U praksi to znači da za sortiranje niza od* **milion** *elemenata sa Merge sort treba ispod* 1 **min.**, *a sa Bubble sort preko* 60 **sati**.