

SPA1 Predavanje 1

- ALGORITMI: želimo analizirati njihovu složenost, tj. koliko resursa računala koriste, ta mislimo na memoriju i vrijeme
- formalno pokazati da rade ono što trebaju raditi (KOREKTNOST)
- STRUKTURA PODATAKA: (data structure)
 - organizirati podatke na način koji omogućava brz pristup podacima
 - npr. polje, lista

PROBLEM SORTIRANJA:

- ulaz: niz elemenata $A = \langle a_1, \dots, a_n \rangle$ i relacija ' \leq ' potpunog uređenja nad elementima niza A
- izlaz: permutacija niza $\langle a'_1, a'_2, \dots, a'_n \rangle$ t.d. je $a'_1 \leq a'_2 \leq \dots \leq a'_n$

Algoritmi za sortiranje:

- algoritmi koji uspoređuju susjedne elemente
npr. Bubble sort, Insertion sort, ...
- brži u praksi:
npr. Merge Sort, Q Sort, Heap Sort, ...

Insertion Sort algoritam

motivacija: slaganje karata u ruci

Primjer: 8 2 4 9 3 6
 ^{KEY}
 2 8 4 9 3 6
 ^{insert}
 2 8 9 3 6
 ^{insert}

2 4 8 9 3 6

2 4 8 9 3 6

2 3 4 8 9 6 \Rightarrow 2 3 4 6 8 9 \checkmark

Sljedeći zadatak je: napisati pseudokod

Koristit ćemo polje $A[1, \dots, n]$ kao strukturu podataka
INDEXI

Insertion Sort (A, n)

for $j \leftarrow 2$ to n

key $\leftarrow A[j]$

$i \leftarrow j-1$

while $i > 0$ and $A[i] > \text{key}$

$A[i+1] \leftarrow A[i]$

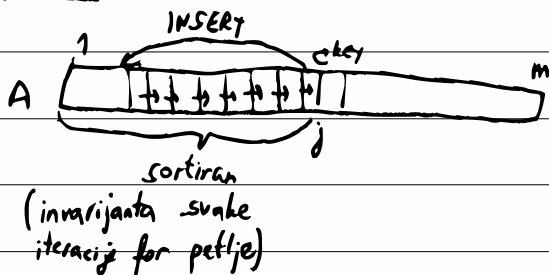
$i \leftarrow i-1$

$A[i+1] \leftarrow \text{key}$

INSERT

Analiza algoritma:

Korektnost:



Formalni argument: (mat. indukcija)

BAZA: $j=2$

PRETPOSTAVKA: $j=k$

KORAK: $j=k+1$

Vremenska složenost algoritma: (VSA) - running time

$$T(n) = \underline{5 \text{ sec}} ???$$

↑

dužina niza

- izraziti VSA je petivijalna stvar (dužina ulaznog niza, neki nizovi "lakši" za sortiranje)

Koristimo u analizi 3 načina izlažavanja VSA

1. "WORST CASE" (najčešći) - daje gornju meću na VSA (generacija)

$t(m)$: maksimalno vrijeme trajanja alg. na inputu dužine m

2. "AVERAGE-CASE" (ponekad)

$T(n)$ = očekivano (prosječno) vrijeme trajanja algoritma na inputu dužine n

Napomena: Pretpostavka da input dolazi iz određene statističke distribucije (vjerojatnost inputa)

3. "BEST-CASE"

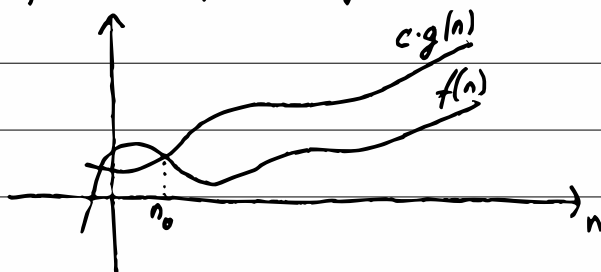
$T(n)$ = minimalno trajanje alg. na inputu dužine n

Naš cilj: Izraziti VSA neovisno o vrsti računala
("IGNORE MACHINE DEPENDANT CONSTANTS")

Ideja: Promatrati $T(n)$ kada $n \rightarrow \infty$

Asimptotska analiza:

- asimptotska notacija
- O -notacija (veliko O) - gornja meda
- pisano $f(n) \stackrel{\text{asimptotska}}{=} O(g(n))$ ako postoje konstante $c > 0$ i $n_0 > 0$ t.d. vrijedi: $0 \leq f(n) \leq c \cdot g(n)$ za sve $n \geq n_0$



npv.: $2 \cdot n^2 = O(n^3)$

$$c = 2, n_0 = 1$$

$$0 \leq 2n^2 \leq c \cdot n^3$$

- definicija veliko O notacije preko skupa:

$$O(g(n)) = \{f(n) \mid \exists c > 0, n_0 > 0 \text{ t.d. } 0 \leq f(n) \leq c \cdot g(n)\}$$

$$f(n) = O(g(n))$$

Ponekad ćemo pisati:

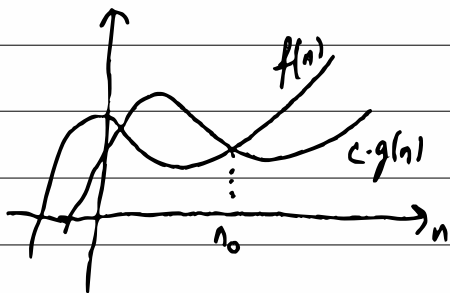
$$f(n) = n^3 + O(n^2) \text{ zapis interpretirajte kao}$$

$$f(n) = n^3 + h(n), \text{ gdje je } h(n) \in O(n^2)$$

Veliko Ω - notacija (donja meta)

$f(n) = \Omega(g(n))$ ako postoje konstante $c > 0$ i $n_0 > 0$ t.d.

$$0 \leq c \cdot g(n) \leq f(n), \text{ za } \forall n \geq n_0$$



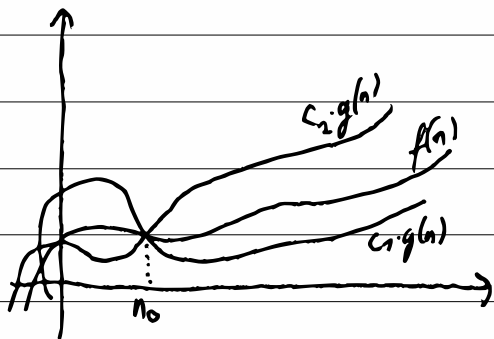
pr.: $\sqrt{n} = \Omega(\log n)$
 $(c=1, n_0=16)$

DZ.: Napišite def. Ω preko skupa:

$\Omega(g(n)) = \{f(n) \mid \exists c > 0, n_0 > 0 \text{ t.d. } 0 \leq c \cdot g(n) \leq f(n)\}$

- velika Θ notacija
 THETA

$f(n) = \Theta(g(n))$ ako postoje konstante $c_1 > 0, c_2 > 0, n_0 > 0$ t.d.
 $0 \leq c_1 \cdot g(n) \leq f(n) \leq c_2 \cdot g(n)$ za sve $n \geq n_0$



SKUPOVNA: $\Theta(g(n)) = O(g(n)) \cap \Omega(g(n))$

pr.: $\frac{1}{2}n^2 - 2n = \Theta(n^2) \left| \begin{array}{l} c_1 = \\ c_2 = \end{array} \right. , n_0 = \right|_{DZ}$

$$\text{npr.: } 3n^3 + 90n^2 - 5n + 6046 = \ominus(n^3)$$

↑
ISPUSTITI ČLANOVE
NIŽEG REDA I IGNOIRATI,
VODEĆU KONST.

$(n \rightarrow \infty)$

VSA za Insertion Sort

"WORST CASE" anal. za (obrnuto sortirani niz)

Insertion Sort (A, n)	CIVENA	BR. PONAVLJANJA
for $j \leftarrow 2$ to n	C_1	n
$\text{key} \leftarrow A[j]$	C_2	$n-1$
$i \leftarrow j-1$	C_3	$n-1$
while $i > 0$ and $A[i] > \text{key}$	C_4	j
$A[i+1] \leftarrow A[i]$	C_5	$j-1$
$i \leftarrow i-1$	C_6	$j-1$
$A[i+1] \leftarrow \text{key}$	C_7	$n-1$

$$T(n) = C_1 n + C_2 \cdot (n-1) + C_3 (n-1) + C_4 \underbrace{\sum_{j=2}^n j}_{\frac{n(n+1)}{2} - 1} + C_5 \underbrace{\sum_{j=2}^n (j-1)}_{\frac{n(n-1)}{2}} + C_6 \underbrace{\sum_{j=2}^n (j-1)}_{\frac{n(n-1)}{2}} + C_7 \cdot (n-1) = \dots$$



$$T(n) = \underbrace{\left(\frac{c_4 + c_5 + c_6}{2}\right)}_{\text{konst.}} \cdot n^2 + \underbrace{\left(c_1 + c_2 + c_3 + \frac{c_4 - c_5 - c_6}{2} + c_7\right)}_{\text{konst.}} \cdot n - \underbrace{(c_2 + c_3 + c_4 + c_7)}_{\text{konst.}}$$

$$\Rightarrow T(n) = \Theta(n^2)$$

DZ: Napravite "best case" analizu za Insertion Sort

Prostorna složenost Insertion Sort alga:

- tzv. "in place" alg.
- prostorna složenost: $\Theta(n)$

RAM (random access machine)

- jednoprocesni model
- instrukcije se izvršavaju slijedno (no concurrency)
- vrijeme potrebno za instrukciju je konstantno

Što su instrukcije u RAM modelu?

- aritmetičke instrukcije (add, minus, *, division, mod, pod, stropl)
- kontrolne instrukcije (if-else, go to, poziv funkcije, return)
- instrukcije na podatcima (load, store, copy)