

# PROBLEM SORTIRANJA

## Problem:

Ulaz: niz (realnih) brojeva  $\langle a_1, a_2, \dots, a_n \rangle$

Izlaz: permutacija niza  $\langle a_1, a_2, \dots, a_n \rangle$ :

$\langle a_{\pi(1)}, a_{\pi(2)}, \dots, a_{\pi(n)} \rangle$  t.d.

$$a_{\pi(1)} \leq a_{\pi(2)} \leq \dots \leq a_{\pi(n)}$$

Primjer: Zadani su niz brojeva  $\langle 8, 2, 4, 9, 3, 6 \rangle$

Ulaz:  $\langle 8, 2, 4, 9, 3, 6 \rangle$

Izlaz:  $\langle 2, 3, 4, 6, 8, 9 \rangle$

$\langle 8, 2, 4, 9, 3, 6 \rangle$
1 2 3 4 5 6
$\langle 2, 3, 4, 6, 8, 9 \rangle$
$\pi(1) = 2$
$\pi(2) = 5$
$\pi(3) = 3$
$\pi(4) = 6$
$\pi(5) = 1$
$\pi(6) = 4$

- tijekom predavanja razmatramo dva pristupa, tj:  
dva algoritma za rješavanje problema sortiranja:

- \* INSERTION-SORT
- \* MERGE-SORT

## INSERTION SORT ALGORITAM

- ideja: - slagati karte u ruci  
- svakoj karti želimo pronaći odgovarajuću poziciju

## Primjer:

Ulaz:

8 2 4 9 3 6

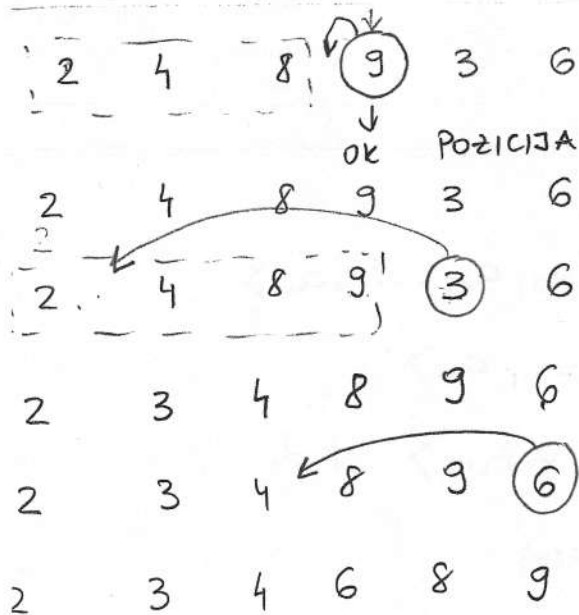
8 2 4 9 3 6

2 8 4 9 3 6

2 8 4 9 3 6

2 4 8 9 3 6

ovdje pretpostavljamo da je podniz koji se sastoji od prvog elementa već sortiran



adatak: Napisati pseudokod INSERTION-SORT ALGORITMA

Upute: koristiti "strukturu podataka" polja A  
(Uvod u računarstvo)

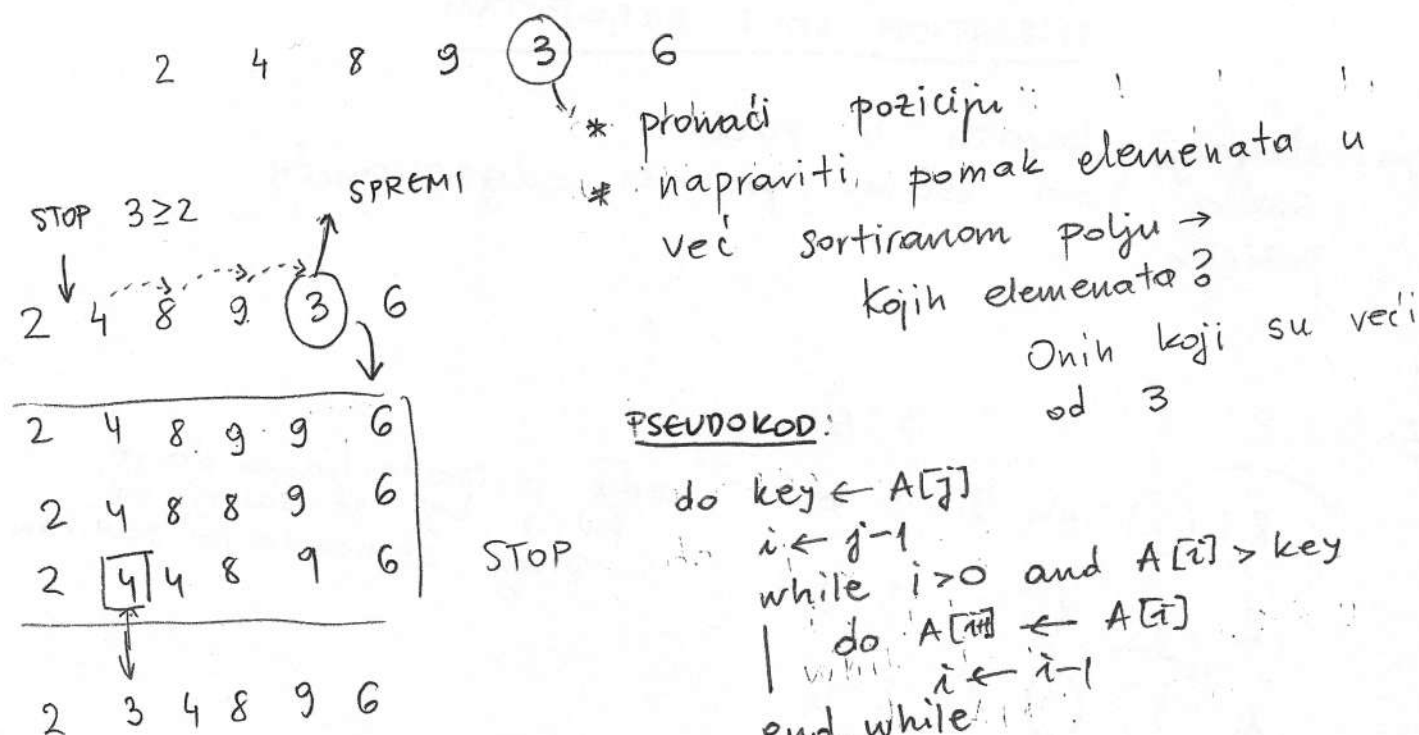
for  $j \leftarrow 2$  to  $\text{length}(A)$

end for

▷ Umetnuti element  $A[j]$  u sortirano polje  $A[1 \dots j-1]$

ovo je samo komentar:  
idemo to implementirati

ogledajmo na jedan od koraka u algoritmu:



kompletni pseudokod:

INSERTION-SORT(A) ↗ dužina polja A, uglavnom se označava s n, tj: length(A) = n

for  $j = 2$  to length(A)

do  $key \leftarrow A[j]$

▷ umetnuti element  $A[j]$  u sortirano polje  $A[1, \dots, j-1]$

$i \leftarrow j-1$

while  $i > 0$  and  $A[i] > key$

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

end while  $i \leftarrow i-1$

$A[i+1] \leftarrow key$

end for

### MERGE-SORT ALGORITHM

- ideja: - podjeli pa vladaj

- rekurzivni pristup
  - podijeli: problem na manje potprobleme
  - vladaj: riješi potprobleme rekurzivno
- ↓  
BOTTOM
- tj: rješavaj sve dok ne dođeš do potproblema koji je dovoljno malen i čije rješenje je trivijalno

ključni MERGE korak ↗

UP ↑

- kombiniraj: rješenja potproblema u rješenje originalnog problema

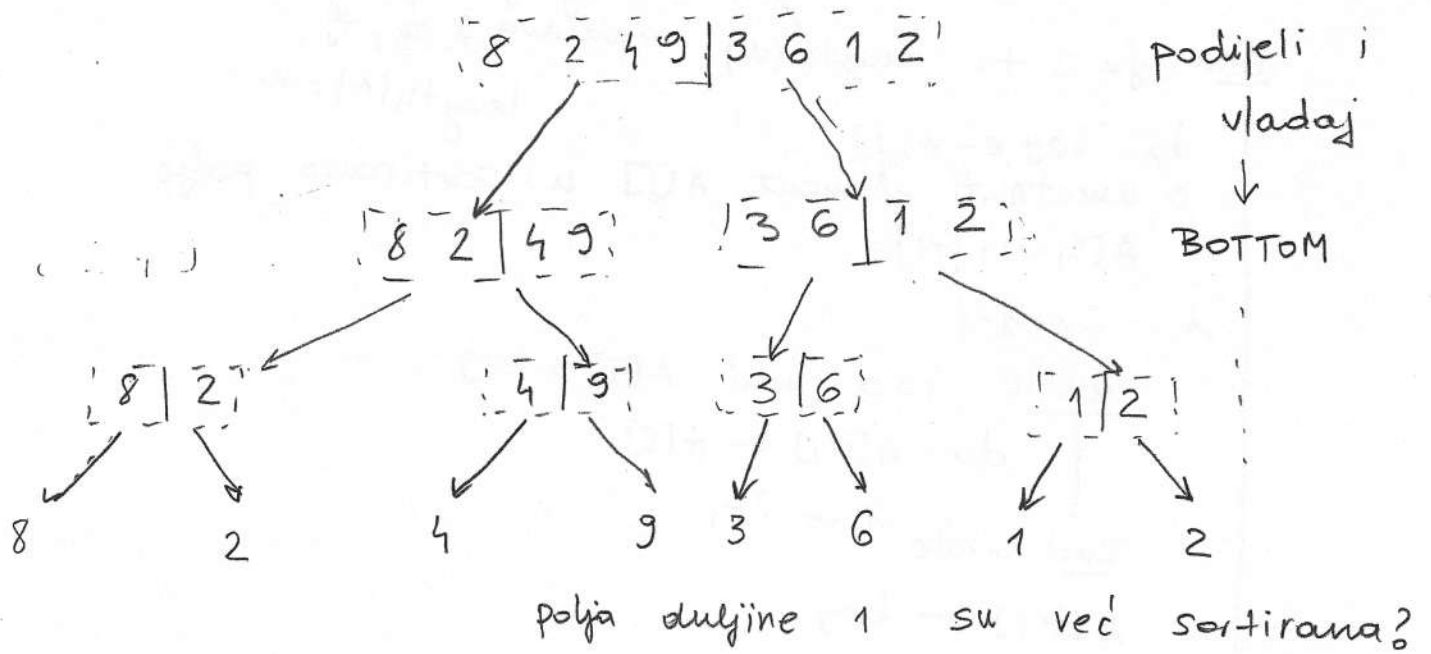
Primjer:

8 2 4 9 3 6 1 2

↖ proširili smo

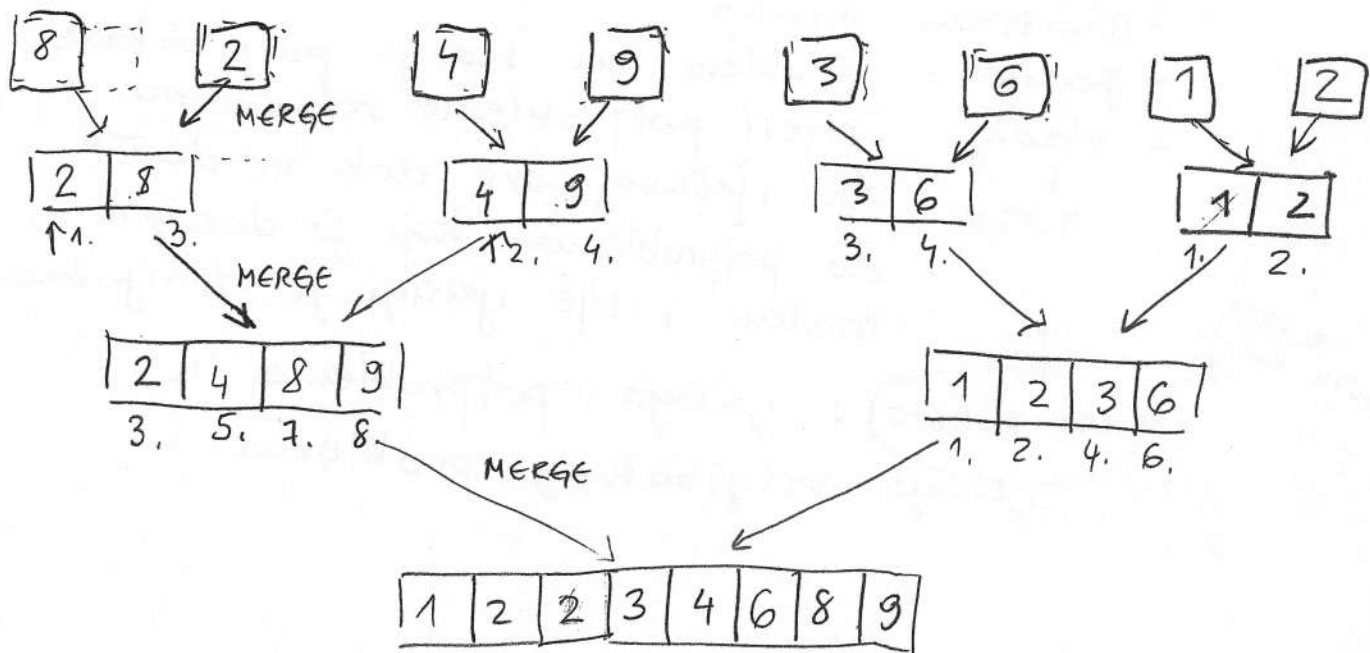
polje iz primjera s početka predavanja za 2 elementa

Što znači zvladati problemom?

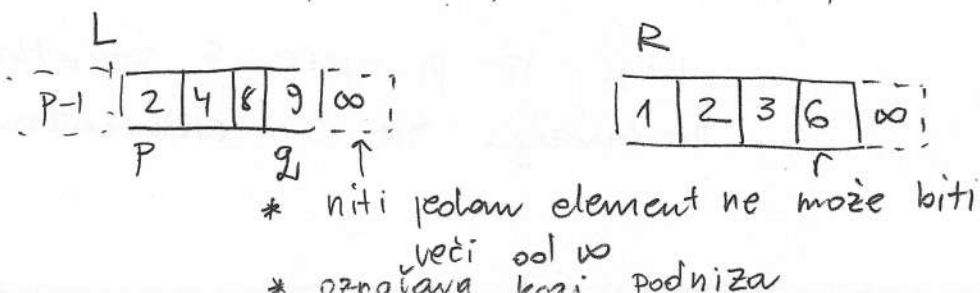


Kako spojiti rješenja potproblema u rješenje originalnog problema?

ključna operacija MERGE



Zadatak: Napisati pseudokod metode MERGE



MERGE (A, p, q, r)

$p \leq q < r$

$n_1 \leftarrow q - p + 1$  du

$n_2 \leftarrow r - q$

▷  $n_1 + 1$  je dužina niza L

▷  $n_2 + 1$  je dužina niza R

▷ L i R su pomoćna polja

II for  $i \leftarrow 1$  to  $n_1$

do  $L[i] \leftarrow A[p + i - 1]$

I for  $j \leftarrow 1$  to  $n_2$

do  $R[j] \leftarrow A[q + j]$

$L[n_1 + 1] \leftarrow \infty$  ← dodatni element ← kad dođemo do  $\infty$   
znamo da je L prazan

$R[n_2 + 1] \leftarrow \infty$

$i \leftarrow 1$

$j \leftarrow 1$

III for  $k \leftarrow p$  to  $r$

do if  $L[i] \leq R[j]$

then  $A[k] \leftarrow L[i]$

$i \leftarrow i + 1$

else  $A[k] \leftarrow R[j]$

$j \leftarrow j + 1$

end if

end for

Zadatak: Napisati pseudokod metode MERGE-SORT



MERGE-SORT ( $p, \underline{r}$ )

if  $p < r$  ← zašto ovaj uvjet?

Kad je  $p \geq r$   
onda smo  
na "dnu rekursije"

then  $q \leftarrow \lfloor \frac{p+r}{2} \rfloor$

MERGE-SORT ( $A, p, q$ )

MERGE-SORT ( $A, q+1, r$ )

MERGE ( $A, p, q, r$ )

end if

## ANALIZA ALGORITAMA

- \* korektnost
- \* vremenska složenost algoritma
- \* (prostorna složenost algoritma)

Korektnost INSERTION-SORT algoritma

↓ LOOP INVARIANT

tema (Invarijanta INSERTION-SORT algoritma)

Na početku svake iteracije for petlje potpolje  $A[1, \dots, j-1]$  sastoji se od prvih  $j-1$  elemenata ulaznog polja  $A$  ali u sortiranom poretku.

Dokaz: matematičkom indukcijom

$j \leftarrow 2$  potpolje  $A[1]$  je sortirano i to je prvi element ulaznog polja  $A$

$j \leftarrow k$  pretpostavka: vrijedi tvrdnja

$j \leftarrow k+1$  korak indukcije:  
nakon umetanja  $A[k+1]$  prvog elementa u  $A[1 \dots k+1]$  postaje  
Sortirano polje  $A[1, \dots, k]$  polje  $A[1 \dots k+1]$  postaje  
Sortirano

za  $j = n$  polje  $A[1, \dots, n]$  postaje sortirano  
Algoritam je korektan

□

# Vremenska složenost INSERTION-SORT ALGORITMA

\* direktna metoda:

\* vrijeme izvršavanja izraziti kao funkciju veličine ulaza

\* u našem slučaju veličinu ulaza "mjerimo" brojem elemenata u polju A

Teorem:

Vrijeme izvršavanja INSERTION-SORT algoritma na polju A duljine n može se opisati funkcijom

$$T(n) = \begin{cases} C_1 n + C_2, & \text{u najboljem slučaju} \\ C_3 n^2 + C_4 n + C_5, & \text{u najgorem slučaju} \\ C_6 n^2 + C_7 n + C_8, & \text{"u prosjeku"} \end{cases}$$

za neke konstante  $C_i \in \mathbb{R}$

INSERTION-SORT

```

1: for j ← 2 to length(A)
2:   do key ← A[j]
3:   i ← j-1
4:   while i > 0 and A[i] > key
5:     do A[i+1] ← A[i]
6:     i ← i-1
7:   end while
8:   A[i+1] ← key

```

cost	times
$C_1$	$n$
$C_2$	$n-1$
$C_3$	$n-1$
$C_4$	$\sum_{j=2}^n t_j$
$C_5$	$\sum_{j=2}^n (t_j-1)$
$C_6$	$\sum_{j=2}^n (t_j-1)$
$C_7$	$n-1$

$t_j \leftarrow$  broj provjera usjeta u while petlji u j-toj iteraciji for petlje  
 $j = 2, \dots, n$

Ukupno vrijeme:

$$T(n) = f_1 n + f_2 (n-1) + f_3 (n-1) + f_4 \sum_{j=2}^n t_j + f_5 \sum_{j=2}^n (t_j-1) + f_6 \sum_{j=2}^n (t_j-1) + f_7 (n-1)$$

vrijeme ovisi o  $t_j$ ,  $j = 2, \dots, n$

\* najbolji slučaj  $t_j = 1$

\* najgori slučaj  $t_j = j$

$j = 2, \dots, n \rightarrow$  polje sortirano

$j = 2, \dots, n \rightarrow$  polje sortirano u 7

\* "u prosjeku" (average case)  $t_j = \frac{j}{2} \quad j = 2, \dots, n$   
 Kad? ako su brojevi slučajno generirani  
 pola manje od  $A[j]$   $\left\{ \begin{array}{l} \text{--} \\ \text{--} \end{array} \right. t_j = \frac{j}{2}$   
 -- veće --

ad Znamo izračunati  $T(n)$ :

z bolji slučaj:

$$\begin{aligned} T(n) &= f_1 n + (f_2 + f_3)(n-1) + f_4(n-1) + f_7(n-1) \\ &= \underbrace{(f_1 + f_2 + f_3 + f_4 + f_7)}_{C_1} n + \underbrace{(-c_2 - c_3 - c_4 - f_7)}_{C_2} \\ &= C_1 n + C_2 \end{aligned}$$

tali slučajevi analogno  $\nabla$

simptotsko ponašanje:

$$T(n) = \begin{cases} \Theta(n) & \text{, najbolji} & \text{worst-case} \\ \Theta(n^2) & \text{, prosječan} & \text{average-case} \\ \Theta(n^2) & \text{, najgori} & \text{best-case} \end{cases}$$

## Korektnost MERGE-SORT ALGORITMA

ma (Invarijanta MERGE algoritma)

Na početku svake iteracije for petlje algoritma MERGE  
 otpolje  $A[p \dots k-1]$  sadrži  $k-p$  najmanjih elemenata iz polja  
 $[1 \dots n_1+1]$  i polja  $R[1 \dots n_2+1]$  u sortiranom poretku.  
 Nadalje,  $L[i]$  i  $R[j]$  su najmanji elementi u poljima  $L$  i  
 koji još nisu kopirani u polje  $A$ .

kaz: Cormen, Leiserson, Rivest, Stein, 2nd ed., 30. str.

rem: (Korektnost MERGE-SORT algoritma)

MERGE-SORT algoritam vraća sortirano polje  $A$ .

z: Iz leme slijedi da  $\text{MERGE}(A, 1, n, \lfloor \frac{n+1}{2} \rfloor)$  vraća  
 sortirano polje  $\nabla$



# Vremenska složenost MERGE-SORT algoritma

MERGE-SORT(A, p, r)

$T(n)$

if  $p < r$

then  $q \leftarrow \lfloor \frac{p+r}{2} \rfloor$

$\rightarrow \Theta(1)$

MERGE-SORT(A, p, q)

MERGE-SORT(A, q+1, r)

MERGE(A, p, q, r)

$\rightarrow T(\lfloor n/2 \rfloor) + T(\lceil n/2 \rceil)$

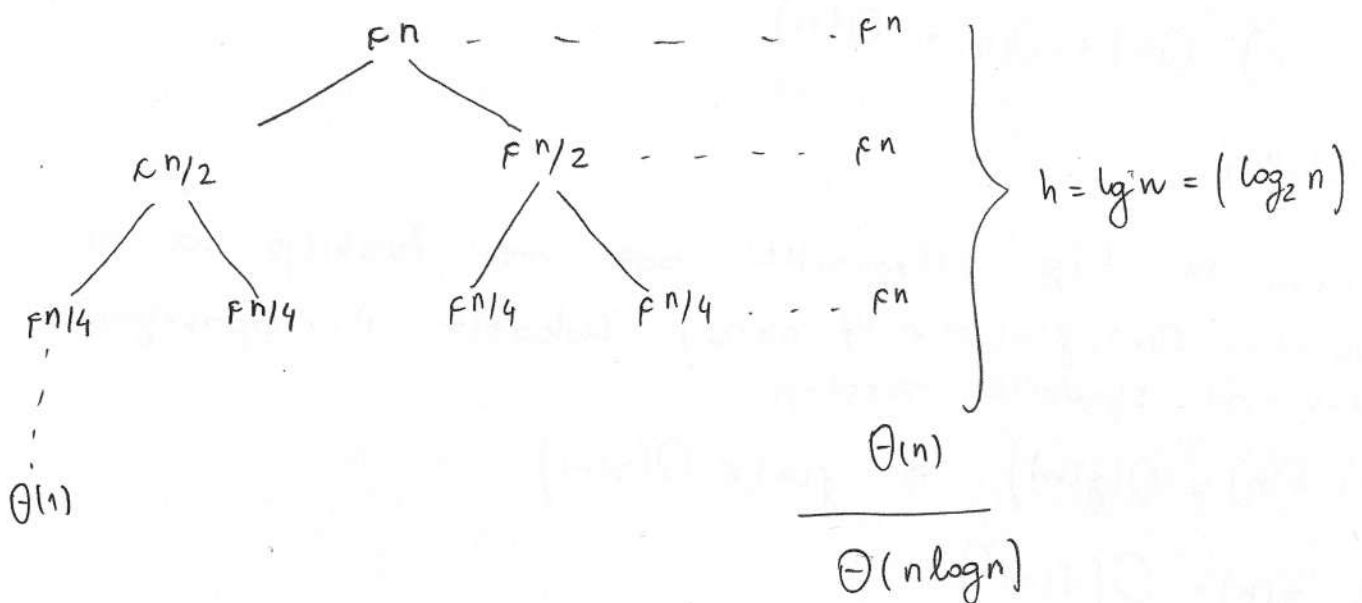
$\rightarrow \Theta(n)$

\* rekurzivna relacija za  $T(n)$

$$T(n) = \begin{cases} \Theta(1), & n=1 \\ 2T(n/2) + \Theta(n), & \text{inače} \end{cases}$$

\* odrediti zatvorenu formulu za  $T(n)$ , tj: iskazati  $T(n)$  kao funkciju od  $n$

\* metoda stabla rekurzije



Teorem MERGE-SORT algoritam vraća sortirano polje u  $\Theta(n \lg n)$  vremenu

skaz: prethodno razmatranje