Predavanje 3

- Potenciranje broja:

Izračunati $a^n, n \in \mathbb{N}$ i a su zadani

NAIVNO:

$$\underbrace{a \cdot a \cdot a \cdot \ldots \cdot a}_{n-1} \quad \text{(for petlja)}$$

$$\Rightarrow \Theta(n)$$

Podijeli-pa-vladaj:

$$a^n = \begin{cases} a^{\frac{n}{2}} \cdot a^{\frac{n}{2}}, & n \text{ paran} \\ a^{\frac{n-1}{2}} \cdot a^{\frac{n-1}{2}} \cdot a, & n \text{ neparan} \end{cases}$$

VSA: $T(n) = 1 \cdot T\left(\frac{n}{2}\right) + \Theta(1) = \Theta(\lg n)$

Fibonaccijevi brojevi:

$$F_n = \begin{cases} 0, & n = 0 \\ 1, & n = 1 \end{cases} \text{ rubni uvjeti}$$
$$F_n = F_{n-1} + F_{n-2}, \quad n > 1 \leftarrow \text{rekurzivni poziv}$$

PSEUDOKOD

```
fib(n)
if n=0 return 0
if n=1 return 1
else return fib(n-1)+fib(n-2)
```

Jeli OVO PODIJELI-PA-VLADAJ ALG?

fib(5)

fib(4) fib(3)

fib(3) fib(2)

Nije PPV jer se potproblemi preklapaju

VSA: $T(n) = T(n-1) + T(n-2) + \Theta(1) = \ldots = \Omega(\phi^n)$

Gdje je $\phi = \frac{1+\sqrt{5}}{2}$ (zlatan rez)

eksponencijalan alg.

$F_0, F_1, F_2, F_3, \ldots, F_n$

računaj brojeve od 0 prema n (for petlja)

$\Rightarrow \Theta(n) = O(n) \cap \Omega(n)$

Možemo li još brže?

TM: $F_n = \frac{\phi^n}{\sqrt{5}}$ ZAOKRUŽEN DO NAJBLIŽEG CIJELOG BROJA

$\Rightarrow T(n)=\Theta(\log n)$ (potenciranje br.)

U praksi nestabilan alg. (ZAOKRUŽIVANJE BROJEVA)

Stabilni algoritam za $F_n$:

TM: $\begin{bmatrix} F_{n+1} & F_n \\ F_n & F_{n-1} \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}^n \Rightarrow \underbrace{T(n)=\Theta(\log n)}_{\text{problem potenciranja}}$

Quick Sort alg. (1962)

- Podijeli-pa-vladaj alg.
- Sortira u mjestu (in-place) poput Insertion Sort alg.
- često se koristi u praksi

Ideja algoritma:

Podijeli polje u dva potpolja oko elementa $x$ (PIVOT) t.d. su elementi u lijevom potpolju $\leq x \leq$ elementi u desnom potpolju

| | | |
|---|---|---|
| $\leq x$ | $x$ | $\geq x$ |

Vladaj: rekurzivno pozovi "Podijeli" na lijevom i desnom potpolju

Kombiniraj: Trivijalan

Ključan korak je "Podijeli" korak

**Pseudo kod:**

Partition $(A, p, q)$ // $A[p \dots q]$

$x = A[p]$ // pivot

$i \leftarrow p$

for $j \leftarrow p+1$ to $q$

  if $A[j] \leq x$

    $i \leftarrow i+1$

    $A[i] \longleftrightarrow A[j]$ // SWAP

$A[p] \longleftrightarrow A[i]$

return $i$ // pozicija pivota

$\boxed{T(n) = \Theta(n)}$

**Primjer:**

$$\begin{array}{cccccccc} p & & & & & & & q \\ 6 & 10 & 13 & 5 & 8 & 3 & 2 & 11 \\ i & j \rightarrow & j \rightarrow & j & & & & \end{array} \quad (x = 6)$$

$i+1 \rightarrow i$

$$\begin{array}{cccccccc} 6 & 5 & 13 & 10 & 8 & 3 & 2 & 11 \\ & i \underset{i+1}{\rightarrow} & i & j \rightarrow & j \rightarrow & j & & \end{array}$$

$$\begin{array}{cccccccc} 6 & 5 & 3 & 10 & 8 & 13 & 2 & 11 \\ & & i \underset{i+1}{\rightarrow} & i & & j \rightarrow & j & \end{array}$$

$$\begin{array}{cccccccc} 6 & 5 & 3 & 2 & 8 & 13 & 10 & 11 \\ & & & i & & & j \rightarrow & j \end{array}$$

$A[p] \leftrightarrow A[i]$

$$\begin{array}{cccccccc} 2 & 5 & 3 & 6 & 8 & 13 & 10 & 11 \\ & & & i & & & & \end{array} \Rightarrow \text{vrati } i$$

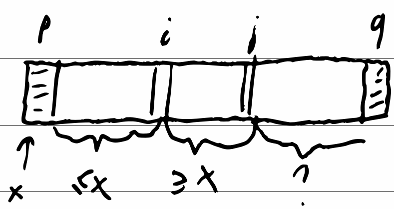
$x =$ pivot

Kolika je VSA Partition na polju $A[1 \dots n]$

$\Rightarrow \underbrace{\Theta(n)}$

prolazite $n$ elemenata i po svakom elementu $\Theta(1)$ posao

Korektnost Partition()




invarijanta iteracije alg.

Pseudokod Quicksort alg.

```
QuickSort(A, p, r)        // A[p..r]
  if p<r
     q ← Partition(A,p,r)
     QuickSort(A, p, q-1)
     QuickSort(A, q+1, r)
```

INICIJALNO ima $A[1 \dots n]$

QuickSort(A, 1, n)

Vremenska složenost Quick Sort

$T(n)$ = "worst case" tj. input $\underbrace{\text{sortiran ili obrnuto sortiran}}$

Partition dijeli $A[1 \dots n]$ na $0 : n-1$

$$\Rightarrow T(n) = \underbrace{T(0)}_{\Theta(1)} + T(n-1) + \underbrace{\Theta(n)}_{\text{podijeli + kombiniraj}} = T(n-1) + \Theta(n)$$

Pr: Stablo rekurzije: $T(n) = T(n-1) + c \cdot n$, za neku konst. $c$

$$T(n) = \begin{array}{c} c \cdot n \\ | \\ T(n-1) \end{array} = \begin{array}{c} c \cdot n \\ | \\ c(n-1) \\ | \\ c(n-2) \end{array} = \ldots = \begin{array}{c} c \cdot n \\ | \\ c(n-1) \\ | \\ c(n-2) \\ \vdots \\ \Theta(1) \end{array} \Rightarrow T(n) = \sum_{i=1}^{n} c \cdot i + \Theta(1)$$

$$= c \cdot \frac{n(n+1)}{2} + \Theta(1) = \Theta(n^2)$$

"BEST CASE": Svaki pivot dijeli polje na dva jednako duga potpolja

$$T(n) = 2 \cdot T\left(\frac{n}{2}\right) + \Theta(n) \overset{MM}{=} \Theta(n \lg n)$$

Čini se da je Quick Sort negdje između $\Theta(n^2)$ i $\Theta(n \lg n)$

Kako odabrati $\underbrace{\text{'dobrog'}}_{\frac{n}{2} : \frac{n}{2}}$ pivota?

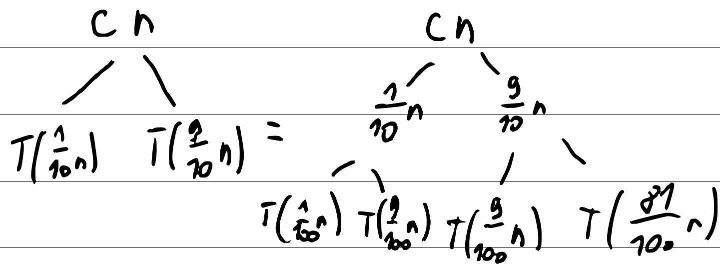
- pretpostavimo da Partition() uvijek dijeli u omjeru $\frac{1}{10} : \frac{9}{10}$

Je li takav pivot dobar ili loš?

VSA: $T(n) = T\left(\frac{1}{10} n\right) + T\left(\frac{9}{10} n\right) + \Theta(n)$

Stablo rekurzije: $T(n) = T\left(\frac{1}{10} n\right) + T\left(\frac{9}{10} n\right) + c \cdot n$, za neki $c$

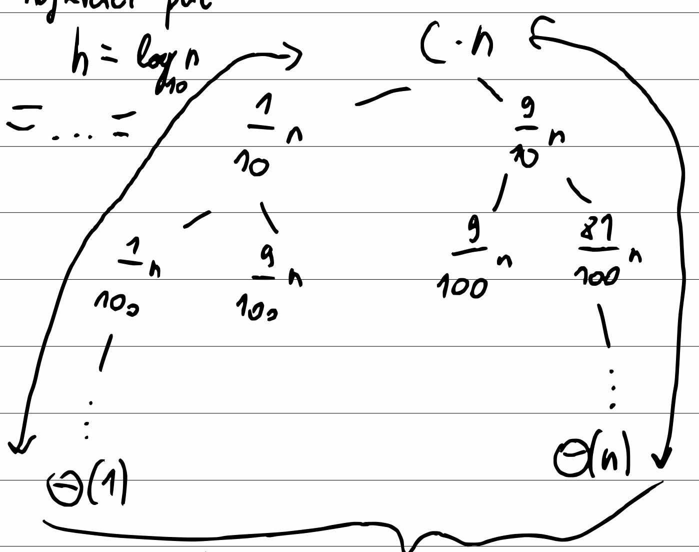
$T(n) =$




najkraći put

$h = \log_{10} n$

$= \ldots =$




NAJDULJI put

$h = \log_{\frac{10}{9}} n$

broj listova je n

- u prvih $\log_{10} n$ nivoa imamo $c \cdot n$ akumulirano, a nakon toga manje od $c \cdot n$

$\Rightarrow \; c \cdot n \cdot \log_{10} n \leq T(n) \leq c \cdot n \cdot \log_{\frac{10}{9}} n + \underbrace{\Theta(n)}_{\text{LISTOVI}}$

$$\boxed{\log_b a = \frac{\log_c b}{\log_c a}}$$

$O(n \log n)$

$\Rightarrow T(n) = \Theta(n \log n)$

$\Rightarrow$ Quick Sort je "brži" $\Theta(n \log n)$ nego $\Theta(n^2)$

Što ako Partition dijeli $\frac{n}{2} : \frac{n}{2}$, pa onda $0 : n-1$ i tako alterira do kraja (dobar pivot pa loš pivot)

$D(n) = 2 \cdot L(n/2) + \Theta(n)$ // $D(n)$ dobar pivot

$L(n) = D(n-1) + \Theta(n)$ // $L(n)$ loš pivot

$$
\begin{aligned}
D(n) &= 2L\left(\frac{n}{2}\right) + \Theta(n) \\
&= 2 \cdot \left(D\left(\frac{n}{2} - 1\right) + \Theta\left(\frac{n}{2}\right)\right) + \Theta(n) \\
&= 2 \cdot D\left(\frac{n}{2} - 1\right) + \Theta(n) = \Theta(n \log n) \quad \text{(MM - slučaj 2)}
\end{aligned}
$$

U praksi:

- odaberi $x$ slučajnim odabirom