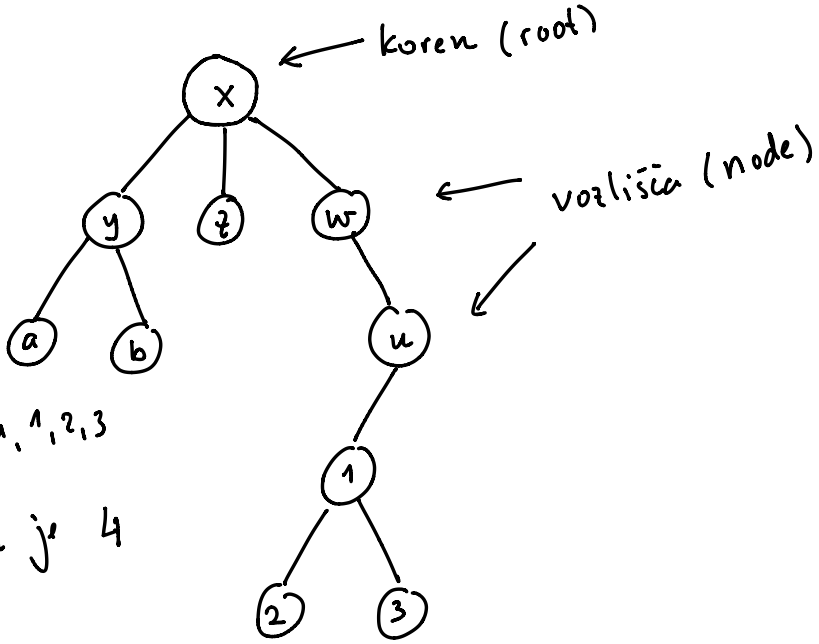
DREVESA



y je oče od a
a je sin y

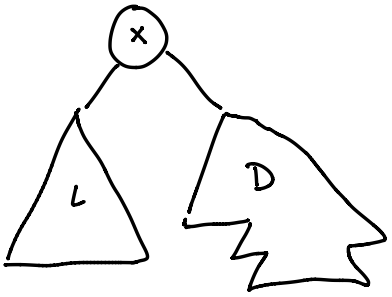
potomci w so u, 1, 2, 3

globina drevesa je 4 (višina)

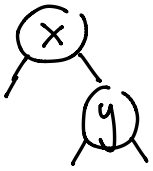
velikost

Dvojiško drevo:

- prazno, ali
- sestavljeno: koren ima dva potomca, ki sta dvojiški drevesi



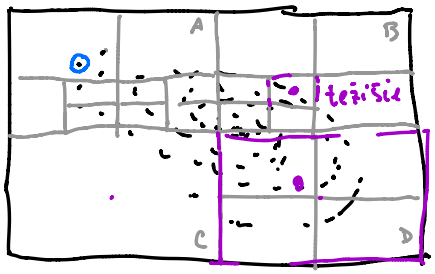
levo in desno poddrevo



Primer: n teles

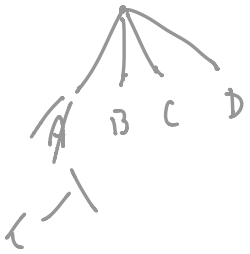
sile: $\frac{n(n-1)}{2}$

$O(n^2)$



(Burnes-Hutt)

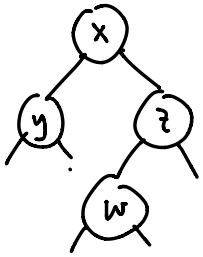
$O(n \cdot \log n)$



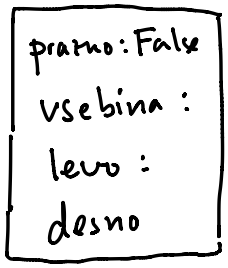
$n = 10^5$

Implementacija:

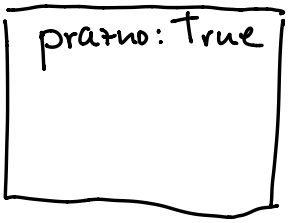
9 objektov (5 ta prazna drevesa)

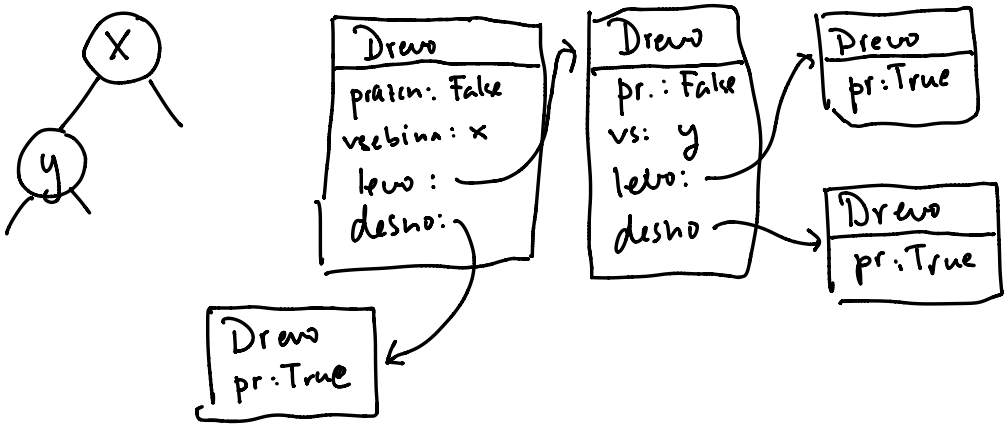


1. možnost:

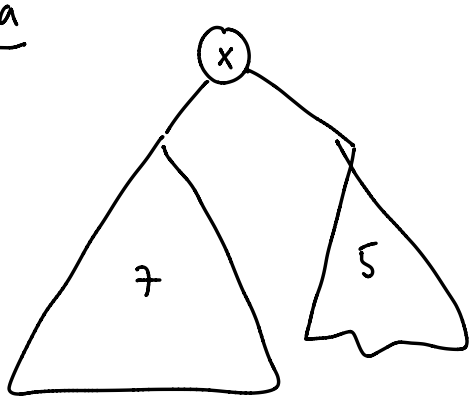


2. možnost

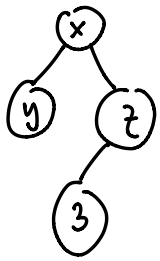




Globina



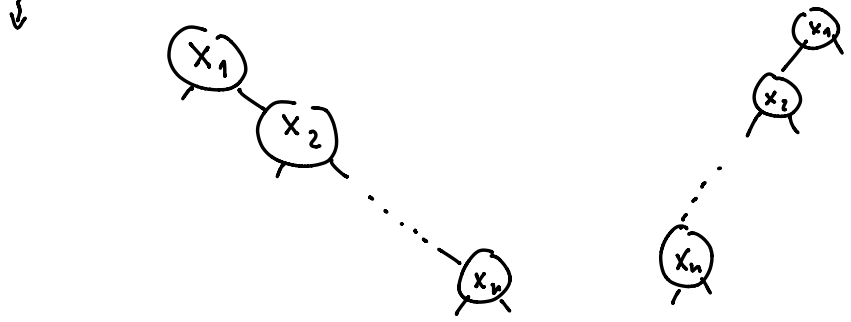
prazen : 0 ⓧ 1



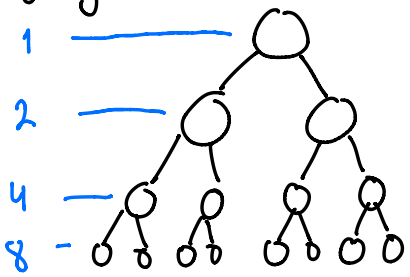
globina(T) =
1 + max(
globina(T.levo),
globina(T.desno))

Posebni primeri:

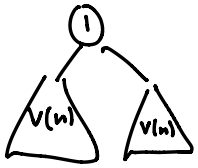
Seznam $[x_1, x_2, \ldots, x_n]$



polno drevo globine n



$n = 4$



globina n → koliko vozlišč?

$$V(0) = 0$$

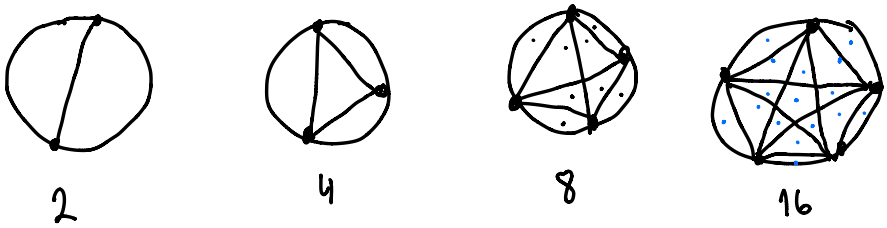
$$V(n+1) = 2 \cdot V(n) + 1$$

n	$v(n)$
0	0
1	1
2	3
3	7
4	15

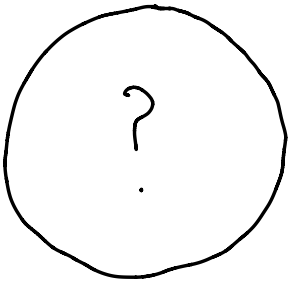
$$V(n) = 1 + 2 + 4 + \dots + 2^{n-1}$$

$$= 2^n - 1$$

Nauk:



2 4 8 16



ISKALNA DREVESA:

Kako narediimo slovar?

To je preslikava iz ključev v vrednosti.

Operacije: 1) poišči vrednost, ki pripada danemu ključu

2) vstavi nov par ključ ↦ vrednost

3) pobriši dani ključ in njegovo vrednost

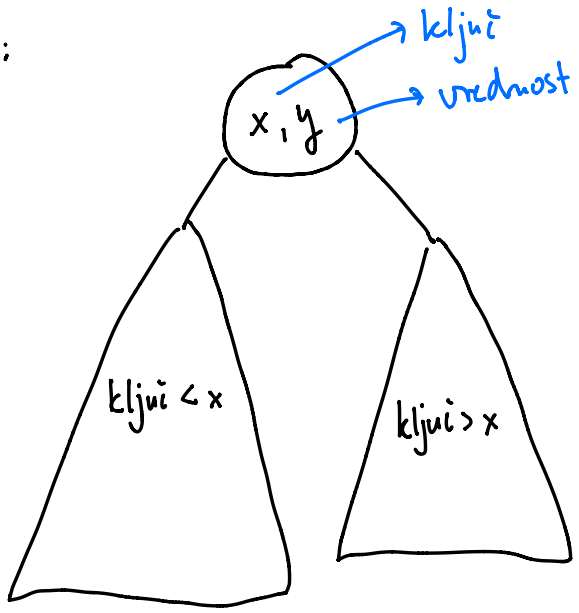
Naivna rešitev: asociativni seznam

Slovar $x_1 \mapsto y_1$, $x_2 \mapsto y_2$, ⋮, $x_n \mapsto y_n$ predstavimo s seznamom parov

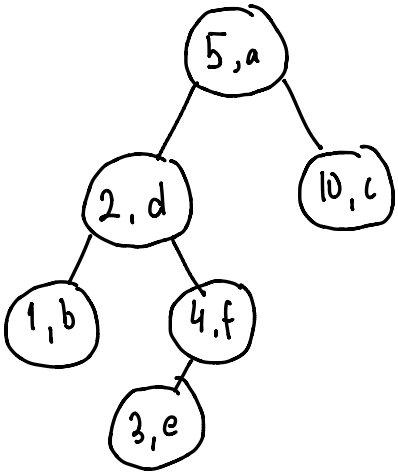
$[(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)]$

1) išči x: $O(n)$
2) vstavi $x \mapsto y$: $O(1)$
3) briši x: $O(n)$

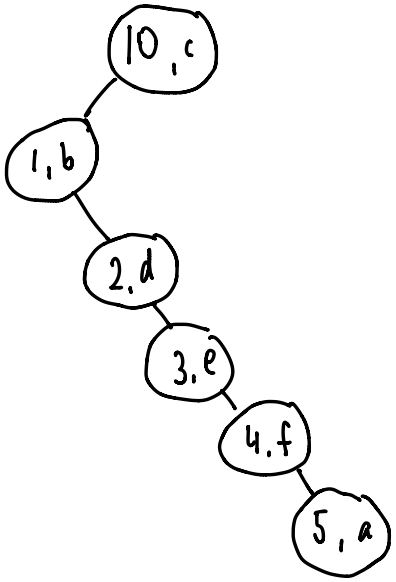
Iskalno drevo:



Primer:
- ✓ 5 → a
- ✓ 2 → d
- ✓ 10 → c
- ✓ 1 → b
- ✓ 3 → e
- ✓ 4 → f



Iščem x



1) Iskanje: iščemo ključ x

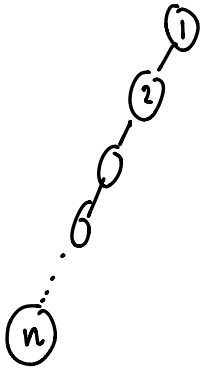
- če je drevo prazno $\Rightarrow$ ni x-a
- če je ključ v korenu $= x \Rightarrow$ smo našli
- če je —||— $< x \Rightarrow$ iščemo v levem poddrevesu
- če je —||— $> x \Rightarrow$ iščemo desno

Časovna zahtevnost v najslabšem primeru:

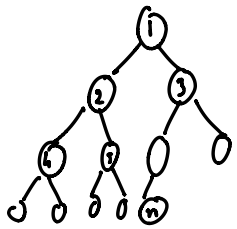
$$O(\text{globina drevesa}) = O(n)$$

vozlišč $= n$

Če ima drevo n vozlišč, kaj vemo o njegovi globini?



$\log_2 n \leq$ globina $\leq n$



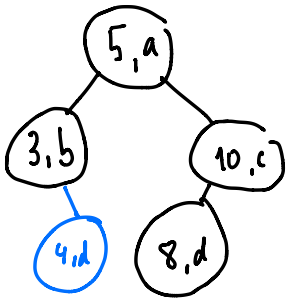
približno $\log_2 n$

polno drevo: $n = 2^{\text{globina}} - 1$

$\text{globina} = \log_2(n+1)$

2) Vstavljanje : vstavi $x \mapsto y$

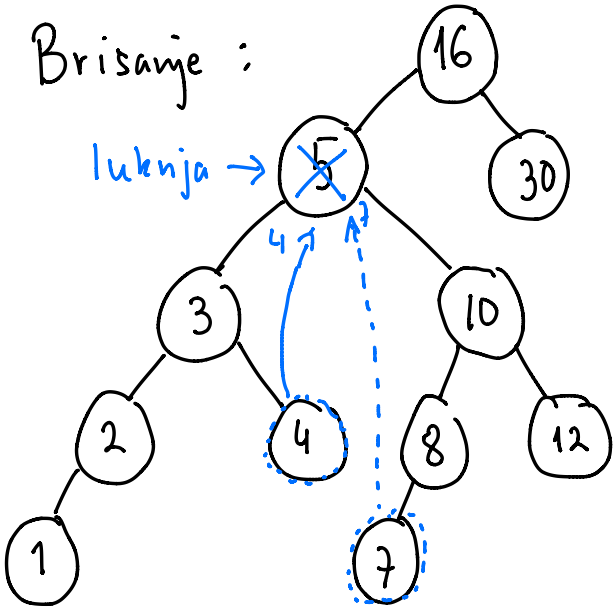
- vstavi $4 \mapsto d$



Vstavimo tja, kjer ga bomo iskali

Zahtevnost : $O(\text{globina}) = O(n)$

3) Brisanje :



pobrišemo 5 :

1) poiščemo 5

2) 5 nadomestimo z največjim v levem ali z najmanjšim v desnem poddrevesu

največji = skrajni desni
najmanjši = skrajni levi

Časovna zahtevnost : $O(\text{globina})$

Dobro iskalno drevo mora imeti čim manjšo globino.