

Dizajn i analiza algoritama

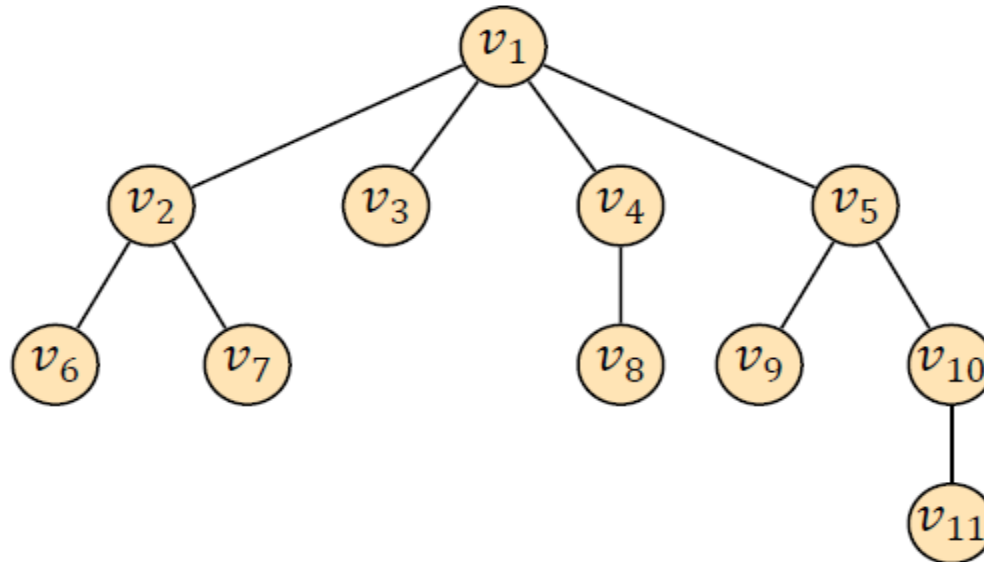
Lekcija 10

leto 2019/2020

Prof. dr Branimir M. Trenkić

Stabla

- Nizovi i liste – odražavaju samo **redosled elemenata**
- **Složeniji odnos** između elemenata
- **Stablo** je **hijerarhijska struktura**
- Korenska stabla (***stabla sa korenom***)
- Primer:

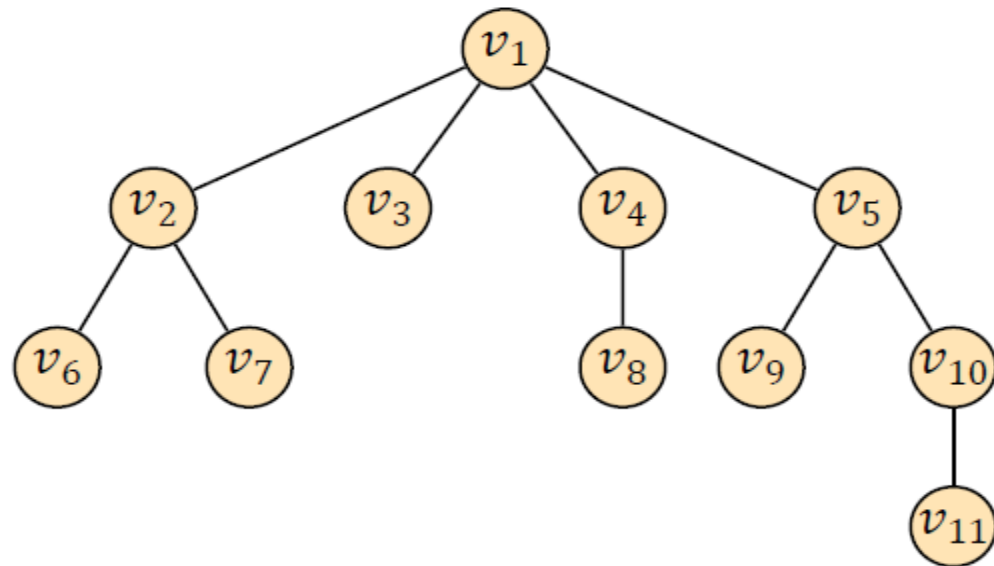


Stabla

- Skup čvorova i skup grana koje povezuju parove različitih čvorova
- **Grana** je uređen **par čvorova**
 - Prvi čvor u paru: **roditelj**
 - Drugi čvor u paru: **dete**
- Jedinstven specijalni čvor na vrhu: **koren**
- **Svaki čvor** osim korena ima jedinstven roditeljski čvor

Stabla

- **Braća (sestre):** čvorovi – **deca istog roditelja**
- **Unutrašnji čvorovi:** imaju **bar jedno dete**
- Spoljašnji čvorovi (**listovi**): **bez dece**
- Primer:
 - **braća:** v_2, v_3, v_4, v_5
 - **listovi:** $v_3, v_6, v_7, v_8, v_9,$
 v_{11}
 - **unutrašnji čvorovi:**
oni koji nisu listovi



Stabla

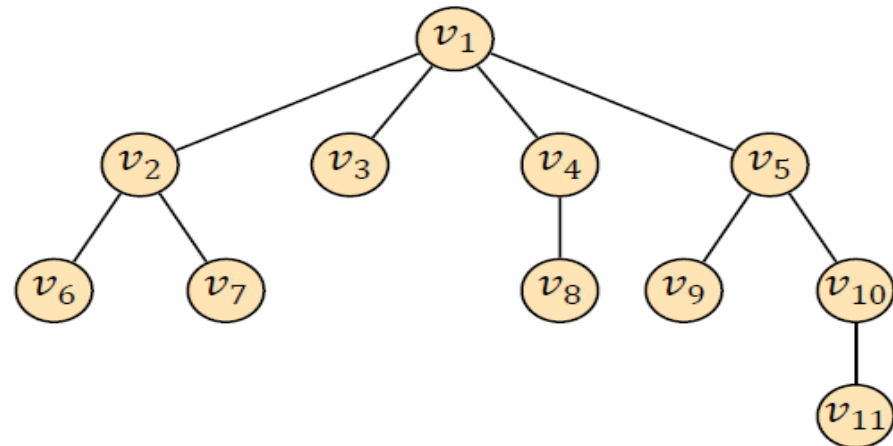
- **Put** između dva različita čvora x i y :
niz čvorova $x = v_1, v_2, \dots, v_m = y$ tako da **postoji grana** između svih susednih čvorova
- **Dužina puta**: **broj grana** na putu
- Konvencija: postoji put od svakog čvora do samog sebe i njegova dužina je 0

- Primer:

- put od v_4 do v_{11} :

v_4, v_5, v_{10}, v_{11}

- dužina tog puta: 4



Stabla

- Postoji **jedinstven put** od svakog **čvora** u stablu **do korena** (i obrnuto)
- **Predak** čvora ***v***: **svaki čvor na jedinstvenom putu** od tog čvora ***v*** do korena
- ***v*** je **potomak** čvora ***u*** ako je ***u*** predak čvora ***v***

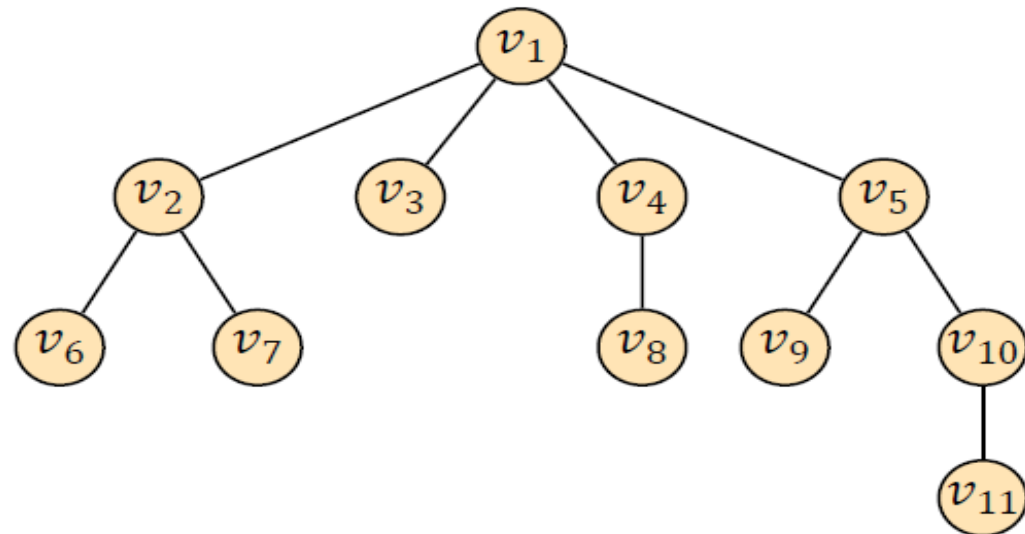
- Primer:

- preci za v_8 :

v_1, v_4, v_8

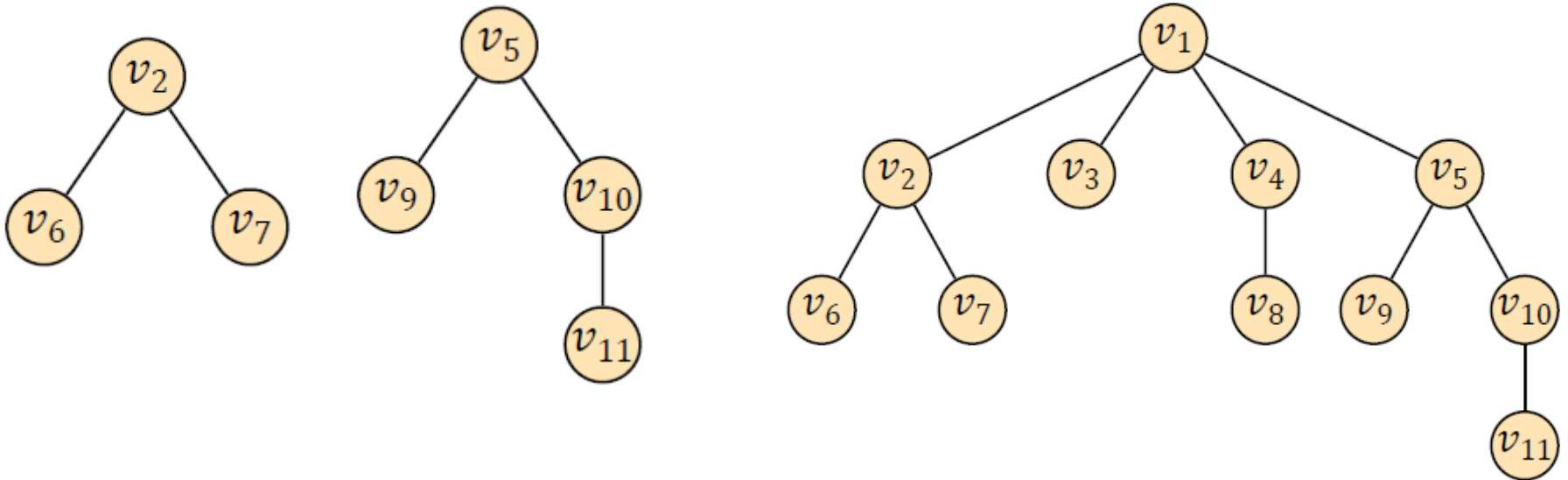
- potomci za v_5 :

v_5, v_9, v_{10}, v_{11}



Stabla

- **Podstablo** sa korenom v : stablo koje se sastoji od čvora v i svih njegovih **potomaka**, zajedno sa nasleđenim granama
- Primer: podstabla sa korenima v_2 i v_5



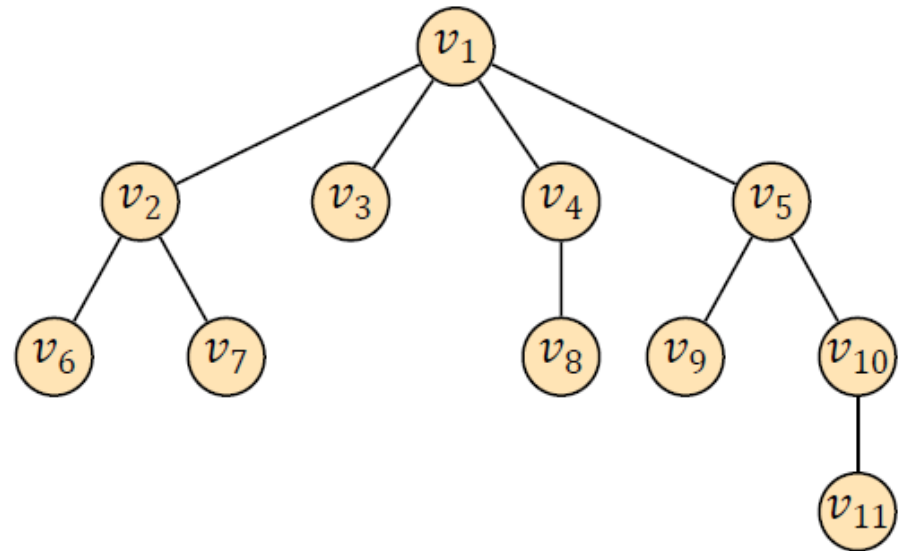
Stabla

- **Visina stabla:** *dužina najdužeg* jedinstvenog *puta* od nekog lista (spoljašnjeg čvora) do korena
- **Nivo stabla:** čvorovi na *istoj dužini puta do korena*

- Primer:

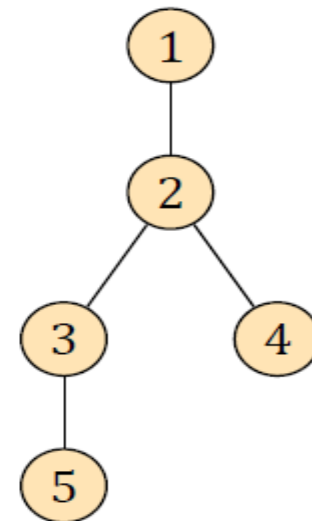
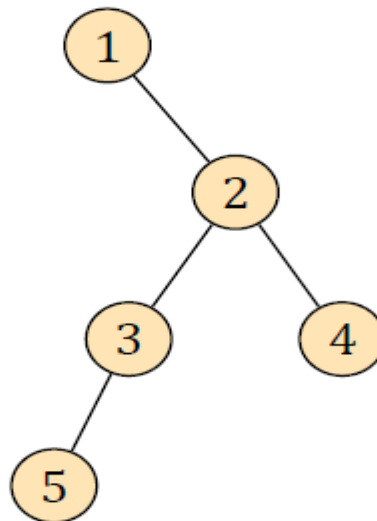
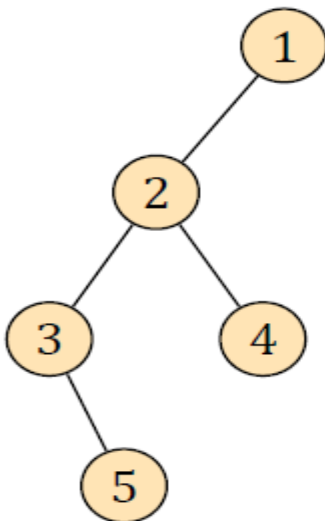
- **visina** stabla: 3

- **nivo 0:** v_1 (koren)
- nivo 1: v_2, v_3, v_4, v_5
- nivo 2: $v_6, v_7, v_8, v_9, v_{10}$
- nivo 3: v_{11}



Binarna stabla

- **Svaki čvor** ima najviše dvoje orijentisane dece (levo ili desno)
- **Čvorovi** binarnog stabla mogu
 1. biti **bez dece**
 2. imati **jedno dete** (levo ili desno)
 3. imati **dvoje dece** (levo i desno)



Binarna stabla

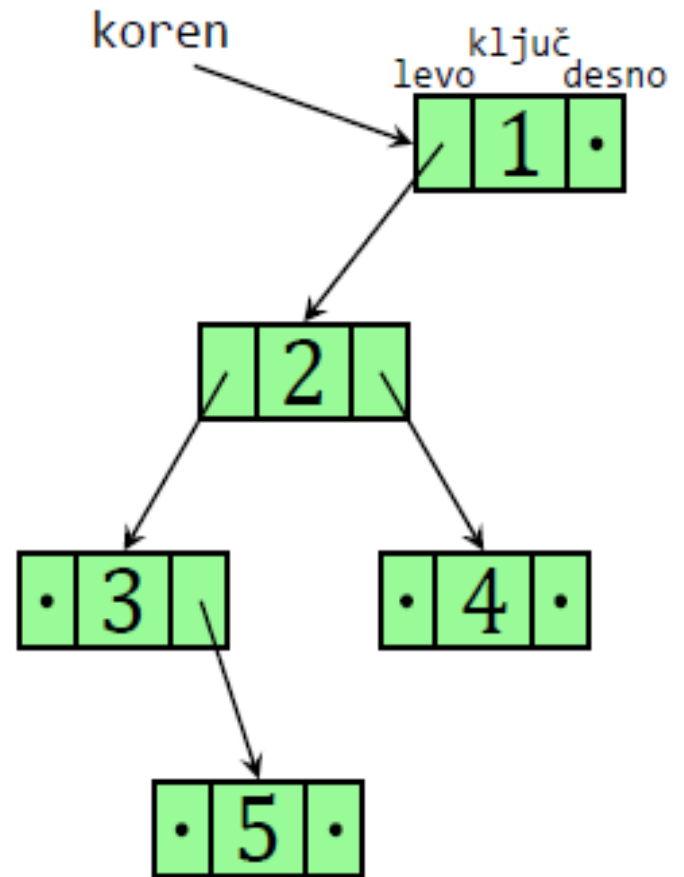
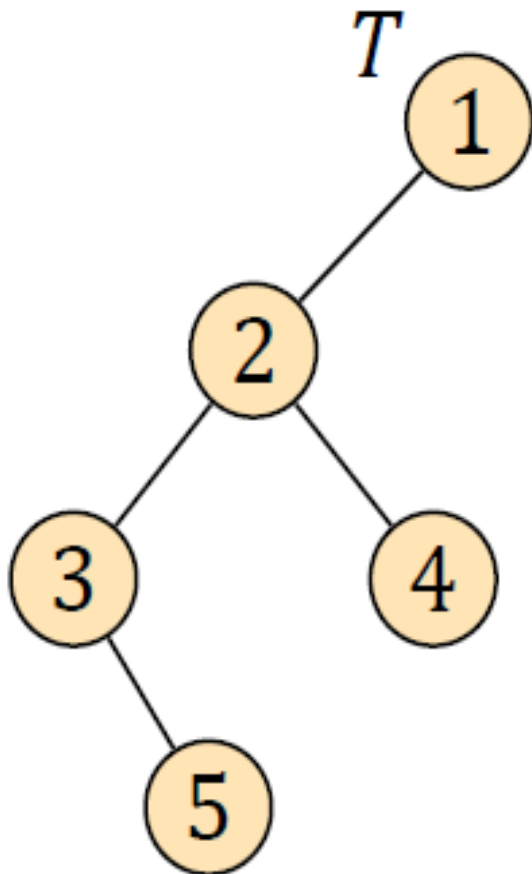
- *Predstavljanje jednog čvora*

```
class Čvor {  
  
    private int ključ;  
    private Čvor levo;  
    private Čvor desno;  
  
    // Konstruktor  
    public Čvor(int k) {  
        ključ = k;  
        levo = null;  
        desno = null;  
    }  
    . . .  
}
```



Binarna stabla

- Primer:*



Operacije nad binarnim stablom

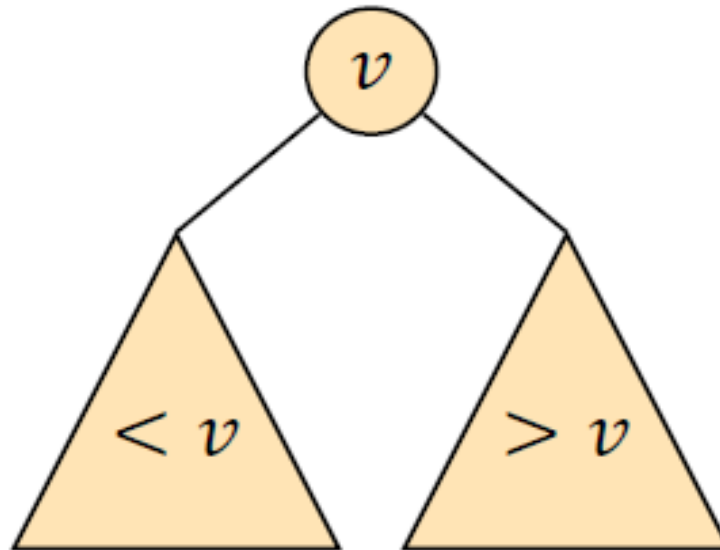
- ***Uobičajene operacije:***
 - Konstruisanje stabla
 - Dodavanje čvora
 - Uklanjanje čvora
 - ...
- ***Posebne operacije:***
 - Nalaženje čvora
 - Obilazak stabla (posećivanje svih čvorova)
 - ...

Operacije nad binarnim stablom

- Za svaku operaciju:
 - 1. *polazna tačka*** je uvek **koren stabla** (jer se *izvan stabla nalazi samo pokazivač na taj čvor stabla*)
 - 2. *u svakom koraku*** se može ***ići samo od čvora do čvora prateći neku granu*** (jer se unutar stabla nalaze pokazivači koji predstavljaju grane između čvorova)

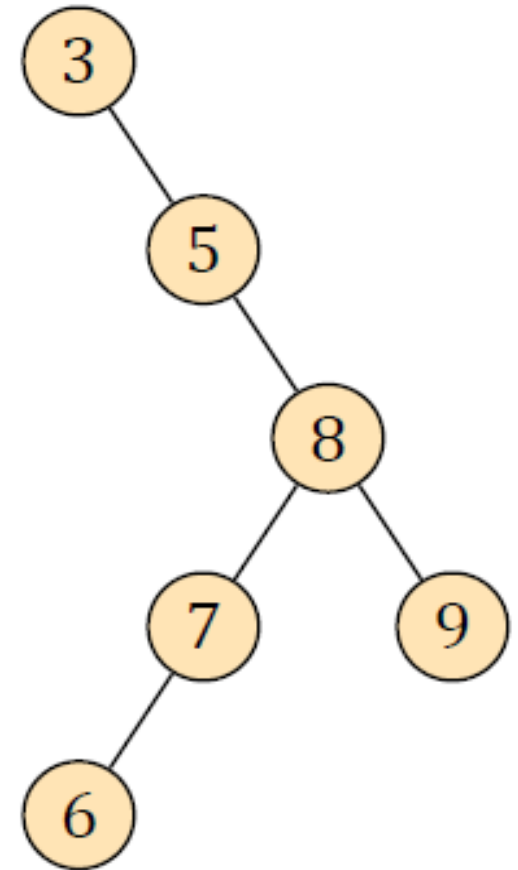
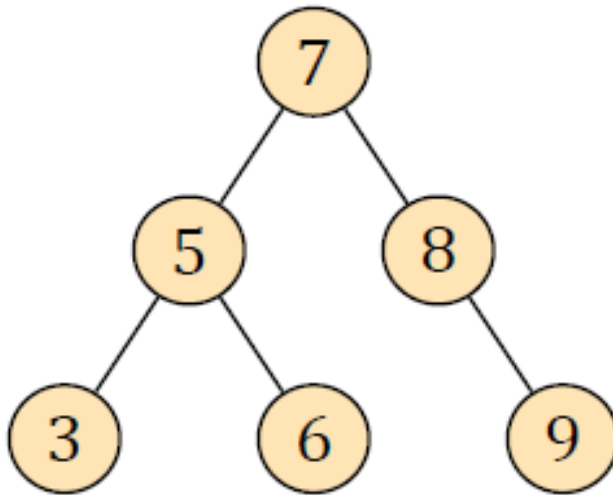
Binarno stablo pretrage

- **BSP** = **binarno stablo** + **BSP svojstvo**
- **BSP svojstvo**: za svaki čvor v ,



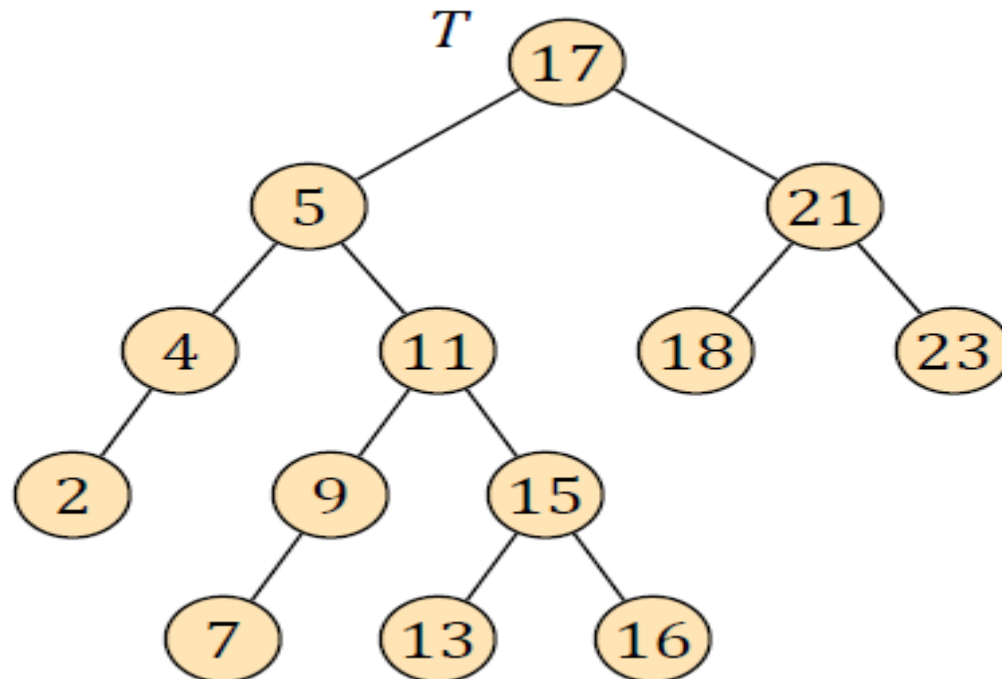
Binarno stablo pretrage

- *Primer*



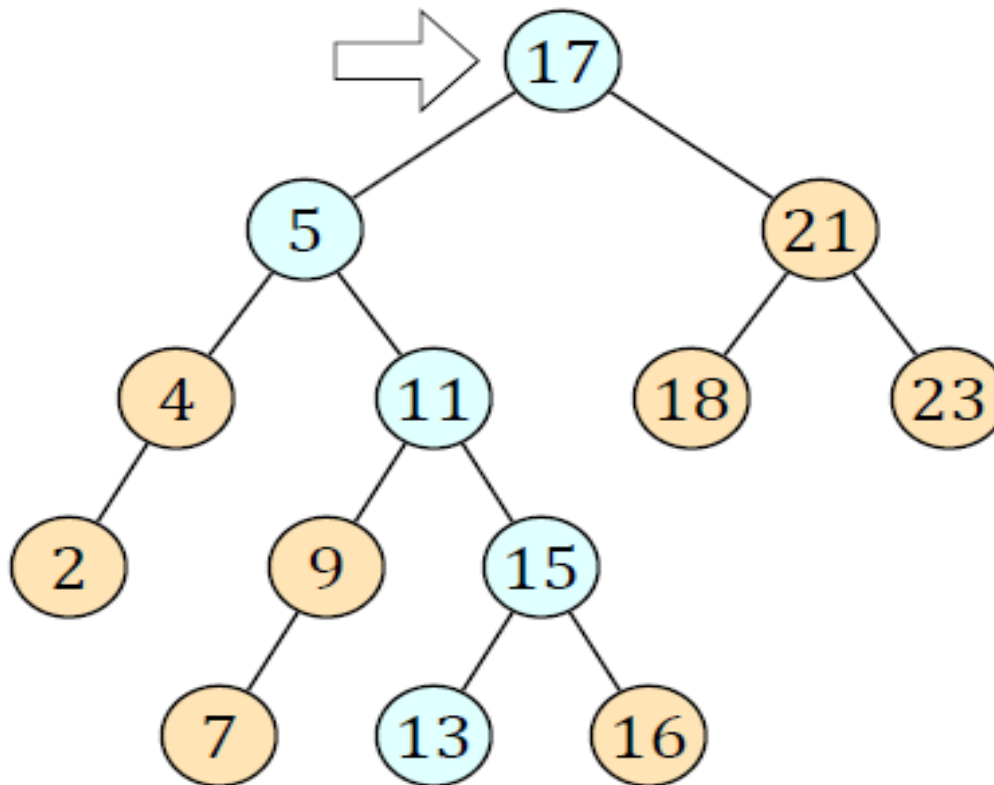
BSP operacije

- Traženje čvora
- za dato BSP T i dati *ključ* x - odrediti da li se čvor sa ključem x nalazi u T
- Primer: traženje čvora **13** u datom BSP-u T



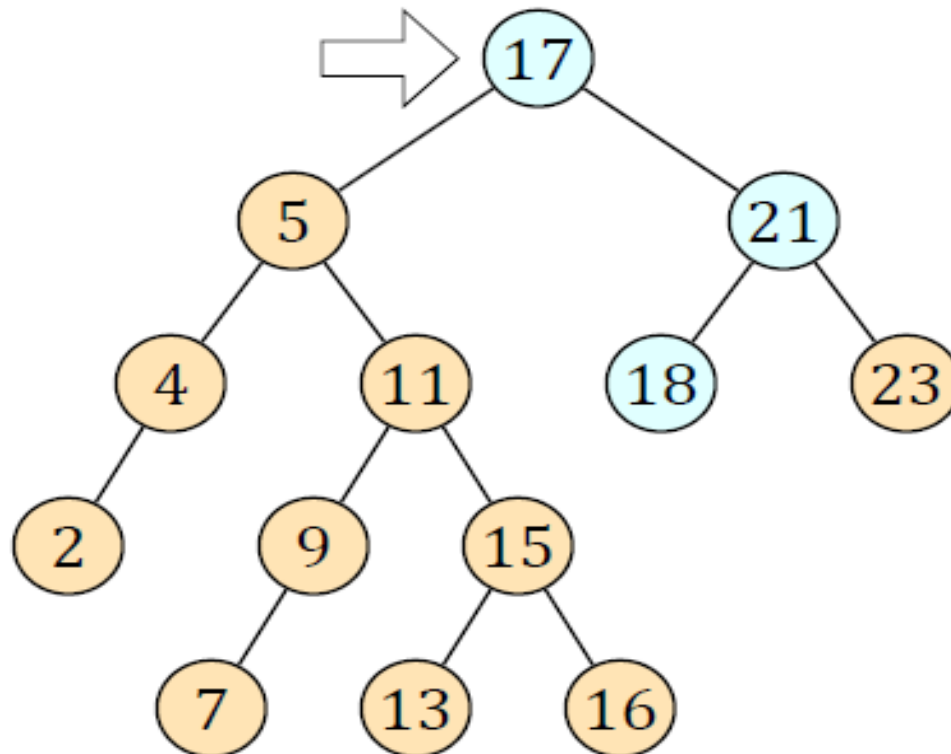
BSP operacije

- Traženje čvora
- Primer: traženje čvora **13** u datom BSP-u T



BSP operacije

- Traženje čvora
- Primer: traženje čvora **20** u datom BSP-u T



BSP operacije

- Traženje čvora - *Algoritam*

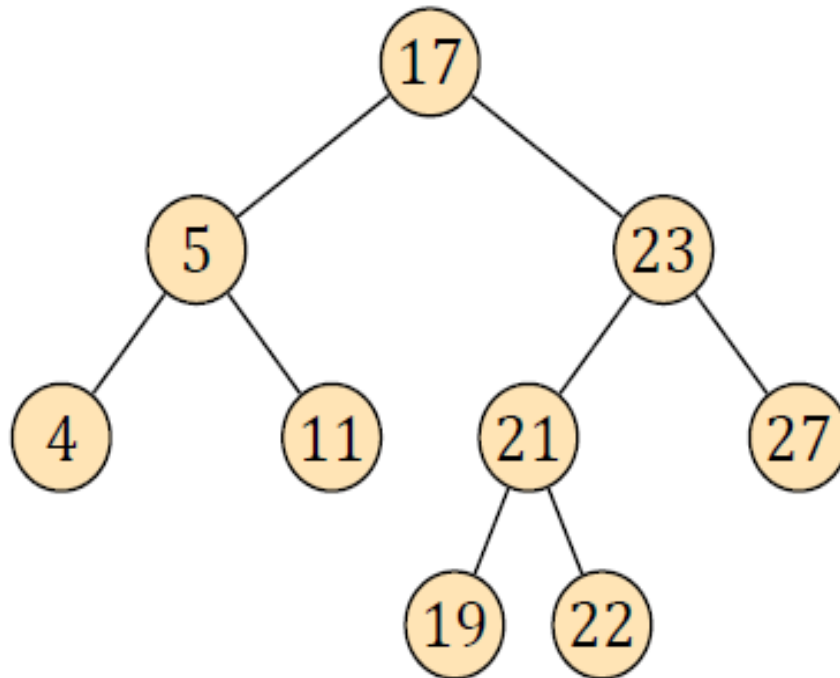
```
// Ulaz: čvor t binarnog stabla pretrage, ključ x
// Izlaz: pokazivač na čvor sa ključem x, ili null
algorithm bst-search(t, x)
```

```
    if ((t == null) || (t.ključ == x)) then
        return t;
    else if (t.ključ > x) then
        bst-search(t.levo, x);
    else
        bst-search(t.desno, x);
```

Vreme izvršavanja: $O(h)$, h – visina stabla

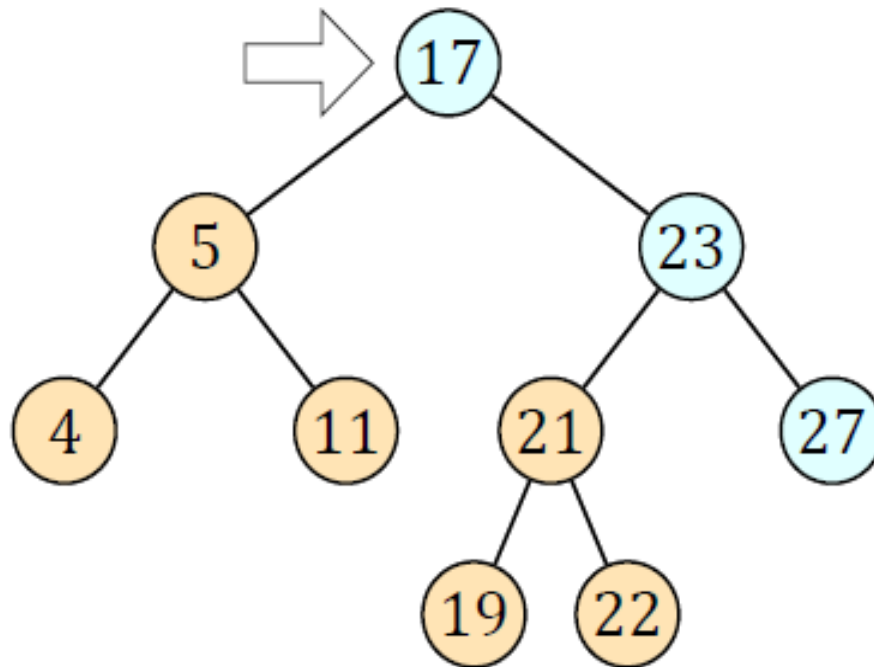
BSP operacije

- **Dodavanje čvora**
- dodati dati čvor **p** u dato BSP **T**
- Primer: dodavanje čvora **25** u datom BSP-u **T**



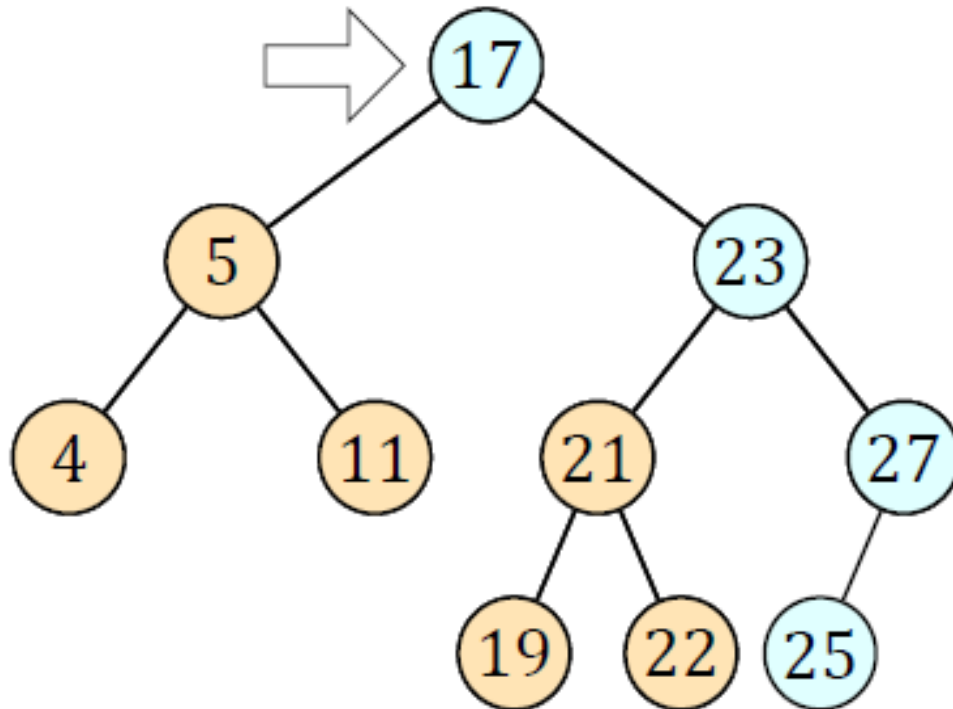
BSP operacije

- **Dodavanje čvora**
- dodati dati čvor **p** u dato BSP **T**
- Primer: dodavanje čvora **25** u datom BSP-u **T**



BSP operacije

- **Dodavanje čvora**
- dodati dati čvor p u dato BSP T
- Primer: dodavanje čvora **25** u datom BSP-u T



BSP operacije

- **Dodavanje čvora – Algoritam**

```
// Ulaz: čvor t binarnog stabla pretrage, novi čvor p
// Izlaz: čvor p dodat u podstablo sa korenom t
algorithm bst-insert(t, p)
```

```
    if (t == null) then
```

```
        t = p;
```

```
    else if (p.ključ < t.ključ) then
```

```
        t.levo = bst-insert(t.levo, p);
```

```
    else if (p.ključ > t.ključ) then
```

```
        t.desno = bst-insert(t.desno, p);
```

```
    else
```

```
        ; // duplikati se ne dodaju
```

```
return t;
```

Vreme izvršavanja: $O(h)$, h – visina stabla