

Stabla

dr Davorka R. Jandrlić dr Goran Lazović

Mašinski fakultet, Univerziteta u Beogradu

Linearne strukture

Nizove i povezane liste održavaju samo redosled elemenata. Često je potrebno predstaviti složeniju vezu među elementima.

Hijerarhijske strukture

Stablo je hijerarhijska struktura, ali može da posluži kao efikasna struktura za neke operacije nad linearnim strukturama.

Korensko stablo čini konačan skup podataka istog tipa koji se nazivaju *čvorovi* i veza među njima (*grana*).

❶ postoji jedan istaknuti čvor koji se naziva **koren stabla**.

2 neka je n čvor stabla i neka su T_1, T_2, \dots, T_k stabla sa korenima n_1, n_2, \dots, n_k redom. Može se konstruisati novo stablo takvo da je n roditelj čvorova n_1, n_2, \dots, n_k . n je tada koren tog stabla, a T_1, T_2, \dots, T_k su njegova podstabla. Za čvorove n_1, n_2, \dots, n_k se kaže da su deca čvora n .

3 Čvorovi povezani sa korenom čine 1. nivo hijerarhije; čvorovi povezani sa njima čine 2 nivo hijerarhije, itd.

④ Svaka grana u stablu povezuje neki čvor sa njegovim (jedinstvenim) prethodnikom (ocem);

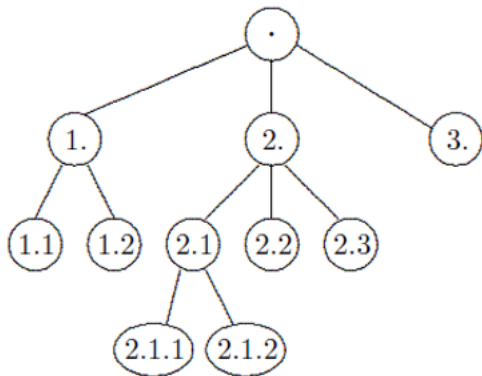
5 Koren je jedini čvor koji nema svog prethodnika.



- 299

1. ...
 1.1. ...
 1.2. ...
2. ...
 2.1. ...
 2.1.1. ...
 2.1.2. ...
 2.2. ...
 2.3. ...
3. ...

```
graph TD; Root((.)) --- 1((1.)); Root --- 2((2.)); Root --- 3((3.)); 1 --- 1.1((1.1)); 1 --- 1.2((1.2)); 2 --- 2.1((2.1)); 2 --- 2.2((2.2)); 2 --- 2.3((2.3)); 2.1 --- 2.1.1((2.1.1)); 2.1 --- 2.1.2((2.1.2));
```



Uređenost čvorova

- Deca čvora su obično uređena sleva u desno.
- Ako eksplicitno želimo da naglasimo da uređenje čvorova nije bitno za stablo ćemo reći da je **neuređeno**.
- Uređenje dece čvorova (braće) sleva u desno može da se proširi na poređenje bilo koja dva čvora koja nisu povezana relacijom predak-potomak na sledeći način:

Ako su a i b braća čvorovi, i čvor a je sa leve strane čvora b , onda su svi potomci čvora a sa leve strane u odnosu na sve potomke čvora b .

Obilazak stabla

- Obilazak stabla predstavlja algoritam kojim se svi čvorovi stabla obilaze tačno jednom.
- Svako obilaženje čvorova stabla predstavlja jedno njihovo uređivanje.
- Najpoznatiji obilasci stabla su PREORDER, INORDER i POSTORDER.
- Neka je T stablo sa korenom r , i neka su T_1, T_2, \dots, T_k podstabla njegovog korena. Tada je:
 - PREORDER – prvo se obilazi koren r , zatim podstablo T_1 u PREORDERu, zatim podstablo T_2 u PREORDERu, ..., podstablo T_k u PREORDERu.
 - INORDER – prvo se obilazi podstablo T_1 u INORDERu, zatim koren r , zatim T_2 u INORDERu, ..., na kraju se obilazi T_k u INORDERu.
 - POSTORDER – prvo se obilazi podstablo T_1 u POSTORDERu, zatim T_2 u POSTORDERu, ..., T_k u POSTORDERu, na kraju se obilazi i koren r .

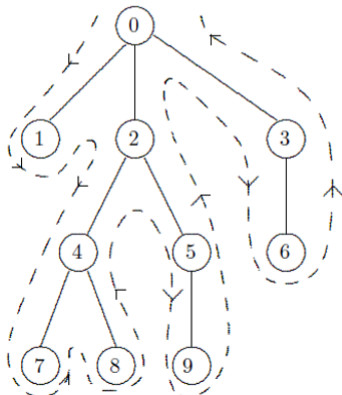
Primer

Obilaženje stabla

PREORDER – 0, 1, 2, 4, 7, 8, 5, 9, 3, 6

INORDER – 1, 0, 7, 4, 8, 2, 9, 5, 6, 3

POSTORDER – 1, 7, 8, 4, 9, 5, 2, 6, 3, 0



Primer

Preorder algoritam u pseudo kôdu

Algoritam **PREORDER**

ulaz: čvor n

izlaz: lista čvorova stabla sa korenom n u PREORDER uređenju

početak

 dodaj n u listu;

 za svako dete c čvora n, ako postoji, s'leva u desno, izvrši

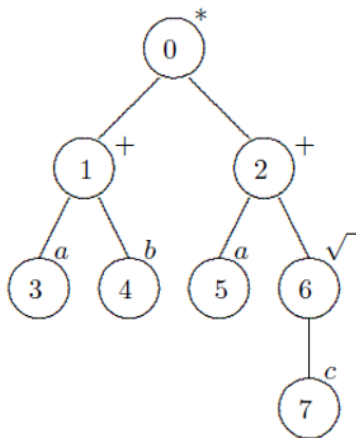
 PREORDER(c);

kraj

Implementacija u C-u je priložena uz materijal sa vežbi

Označena stabla i stabla izraza

Ponekad je korisno pridružiti oznaku i/ili vrednost svakom čvoru stabla, na sličan način kao elementima liste. Oznaka čvora se čuva kao vrednost čvora, i ne predstavlja isto što i njegovo ime.



$$(a + b) * (a + \sqrt{c})$$

Apstraktni tip podataka - Stablo

- **node** - bilo koji tip koji predstavlja naziv čvora. NULL čvor – nepostojeći čvor.
- **labeltype** – bilo koji tip koji predstavlja oznaku pridruženu čvoru. Ukoliko je reč o označenom stablu.
- **tree** – uređeno stablo čiji su čvorovi podaci tipa node. Svakom čvoru je pridružen podatak tipa labeltype.
- Operacije:
 - $\text{ROOT}(T)$ – vraća čvor koji je koren stabla T , ili NULL čvor ukoliko je stablo T prazno.
 - $\text{MAKENULL}(T)$ – prazni stablo T
 - $\text{PARENT}(n, T)$ – vraća roditelj-čvor čvora n u stablu T , NULL čvor ukoliko čvor nema roditelja (ukoliko je koren).
 - $\text{INSERT_CHILD}(l, x, T)$ – u stablo T dodaje čvor sa oznakom l kao prvo po redu dete čvora x .
 - $\text{INSERT_SIBLING}(l, x, T)$ – u stablo T dodaje čvor sa oznakom l kao sledećeg brata čvora x .

- FIRST_CHILD(x, T) – vraća prvo po redu dete čvor x u stablu T , ako postoji, NULL čvor inače.
- NEXT_SIBLING(x, T) – vraća sledeći brat-čvor čvora x u stablu T ako postoji, NULL čvor inače.
- LABEL(x, T) – vraća oznaku čvora x u stablu T .

Napomena: Operacija nad stablima su prilagođene tipu stabla koje se koristi i ne moraju nužno sve biti implementirane.

Predstavlanje stabla

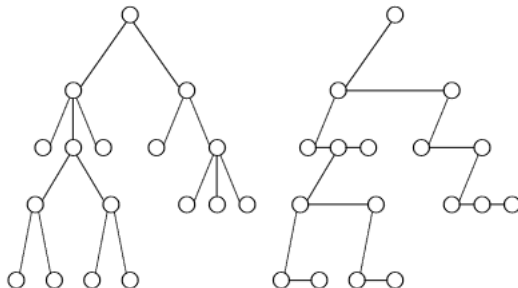
U zavisnosti od toga da li se stablo predstavlja preko pokazivača ili ne, predstava stabla može biti

- **eksplicitna** ili
- **implicitna**.

Stablo stepena većeg od 2 moguće je predstaviti korišćenjem samo dva pokazivača po čvoru tako što:

- prvi pokazuje na prvog sina,
- a drugi na narednog brata.

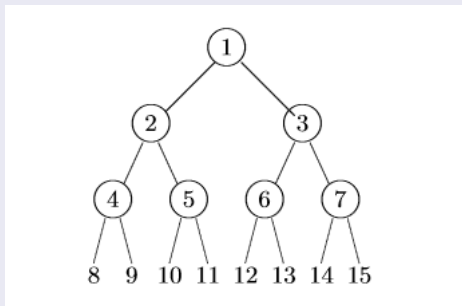
Naredna slika demonstrira prevodjenje stabla (levo) u stablo u kome je svaki čvor predstavljen sa dva pokazivača (desno).



Kod implicitnog predstavljanja stabla, svi ključevi (vrednosti) čvorova se smeštaju u niz i veze među njima proističu iz njihovog mesta u nizu. Npr., binarno stablo se vrlo jednostavno predstavlja nizom tako što se elementi smeštaju u niz $A[1..n]$ na sledeći način:

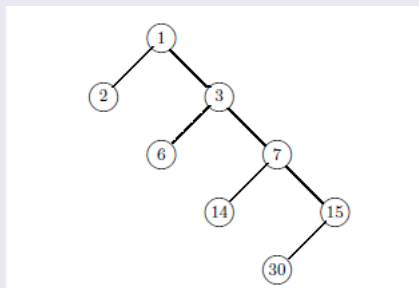
Redosled kojim se čvorovi smeštaju u niz

- Koren se smešta na poziciju $A[1]$,
- levi odnosno desni sin na pozicije $A[2]$ i $A[3]$ itd.
- Zapravo za čvor $A[i]$ sinovi se smeštaju na pozicijama $A[2i]$ i $A[2i + 1]$.



- U nizu se čvorovi prikazuju u onom redosledu u kome se stablo obilazi sleva na desno, nivo po nivo.

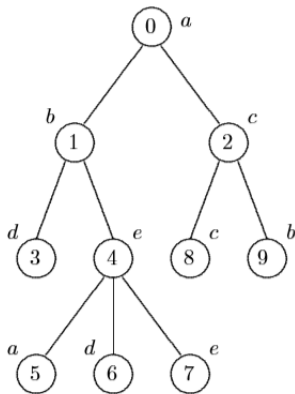
- Mana: mora se rezervisati mesto i za čvorove stabla koji ne postoje. Nije pogodan za neuravnoteženo stablo kakvo je prikazano na slici.



Neophodno je rezervisati 30 mesta (niz dužine 30), iako će većina elemenata niza biti "prazna".

Predstavljanje stabla na osnovu veze "čvor-roditelj"

U primeru je prikazana stablo koje nije binarno, a uz svaki element (u ovom slučaju oznaku-labelu) se čuva i roditelj čvora čime je sačuvana veza čvor-roditelj.



root
0

	label	parent
0	<i>a</i>	-1
1	<i>b</i>	0
2	<i>c</i>	0
3	<i>d</i>	1
4	<i>e</i>	1
5	<i>a</i>	4
6	<i>d</i>	4
7	<i>e</i>	4
8	<i>c</i>	2
9	<i>b</i>	2

Implementacija stabla, predstavljenog na osnovu veze "roditelj-dete", pomoću statičkog niza

- U primeru su nazivi čvorova celi brojevi 0, 1, 2, ..., $n - 1$, gde je n broj čvorova stabla, što ne umanjuje opštost.
- Potrebna je sledeća struktura za opisivanje stabla iz primera:

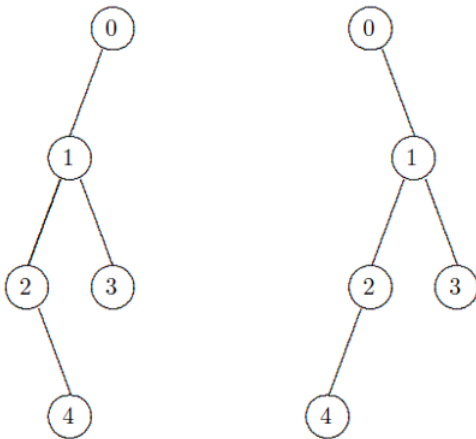
```
// Maksimalni broj cvorova stabla
#define MAX_BROJ_CVOROVA 50

#define NULL_CVOR -1
typedef int cvor; // Tip cvora
typedef char tipoznake; // Tip oznaka pridruženoj svakom
                        // cvoru.

typedef struct {
    cvor koren;
    tipoznake oznake[MAX_BROJ_CVOROVA];
    cvor ocevi[MAX_BROJ_CVOROVA];
}
```

Binarno stablo

Binarno stablo je stablo stepena dva, kod kojeg je bitan redosled dece čvorova, tj. koje dete je levo odnosno desno.

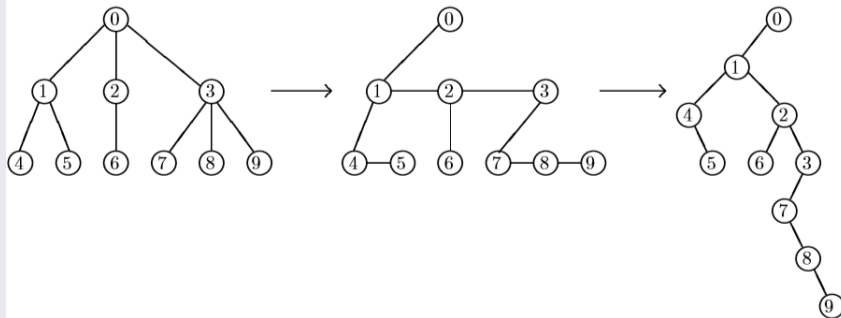


Znakovi su kodirani nizovima bitova. Dekodiranje bi odgovaralo obilasku binarnog stabla na slici.

```

graph TD
    Root(( )) ---|0| L1L(( ))
    Root ---|1| L1R(( ))
    L1L ---|0| L2L(( ))
    L1L ---|1| L2R(( ))
    L1R ---|0| L2RL(( ))
    L2L ---|0| L3L((a))
    L2L ---|1| L3L2((b))
    L2R ---|0| L3R((c))
    L2R ---|1| L3R2((d))
    L2RL ---|0| L3RL((e))
  
```

Interpretacija uređenog stabla kao binarnog



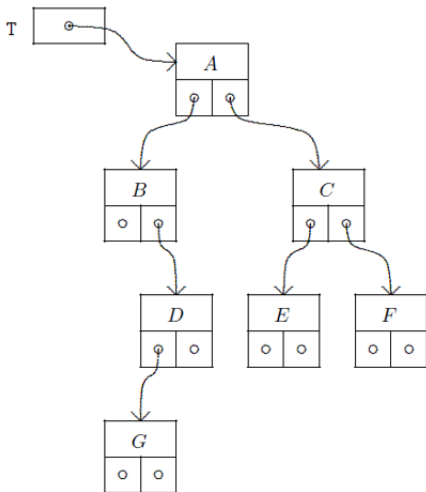
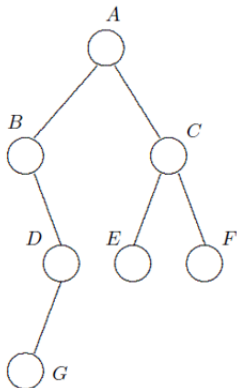
Napomena: svako uređeno stablo može da se prevede u binarno, na osnovu veze "roditelj -levo dete" i "levo dete-brat" (prvi desno). Jasno je da se prevođenjem gubi veza brat-brat, jer u suprotnom prevođenje ne bi bilo moguće ali ostaje zadržano uređenje.

Primer

Struktura potrebna za implementaciju binarnog stabla pomoću pokazivača

```
typedef struct _cvor {  
    char *oznaka; //slučaj kada se čuva samo oznaka  
    struct _cvor *levo;  
    struct _cvor *desno;  
} cvor;  
  
typedef struct cvor* bstablo;
```

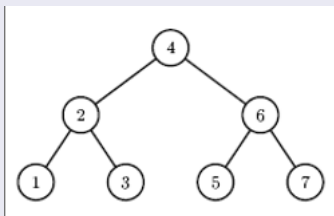
Struktura čvor treba da odgovara onome što će se čuvati u čvoru stabla (npr. može da bude vrlo slična elementu liste). Samo stablo se definiše kao pokazivač na prvi čvor u stablu - koren stabla.



Binarno stablo pretrage

Binarno stablo pretrage je binarno stablo u kojem važi:

- ključ svakog čvora veći je od ključeva svih čvorova levog podstabla, a manji od ključeva svih čvorova desnog podstabla.



- Iz definicije stabla binarne pretrage sledi da se čvor sa najmanjom vrednošću nalazi u najlevljem čvoru. Slično, ključ sa najvećom vrednošću se nalazi u krajnjem desnom čvoru.

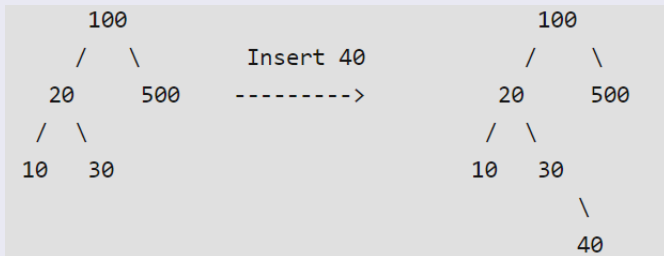
BSP

Osobine:

- I levo i desno podstablo je BST
- Nema dupliranih čvorova
- Obezbeđena uređenost ključeva \Rightarrow efikasna pretraga, pronalaženje minimuma, maximuma

Dodavanje novog čvora

Novi čvor se uvek dodaje kao list u stablu.



Složenost operacija pretrage i dodavanja elementa

Vremenska složenost pronalaženja čvora sa zadatim ključem ili dodavanje odgovarajućeg čvora je $O(h)$, gde je h dubina stabla.

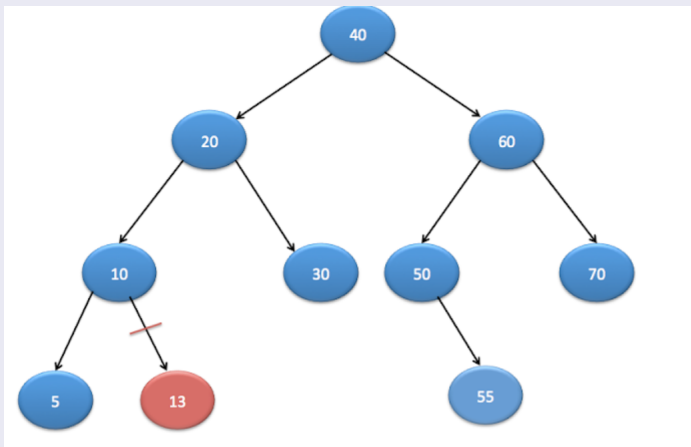
INORDER obilaženjem stabla se uvek dobije sortirani izlaz (sortiran skup ključeva/vrednosti).

Brisanje čvora iz BST

Moguće su sledeće situacije:

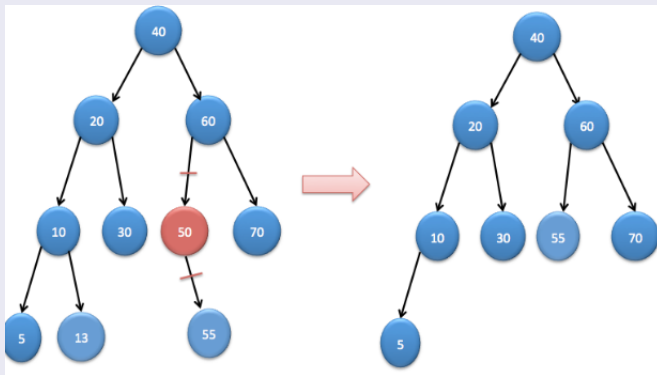
- Čvor koji treba obrisati je list.
- Čvor koji treba obrisati ima samo jedno dete.
- Čvor koji treba obrisati ima 2 deteta.

Brisanje čvora lista



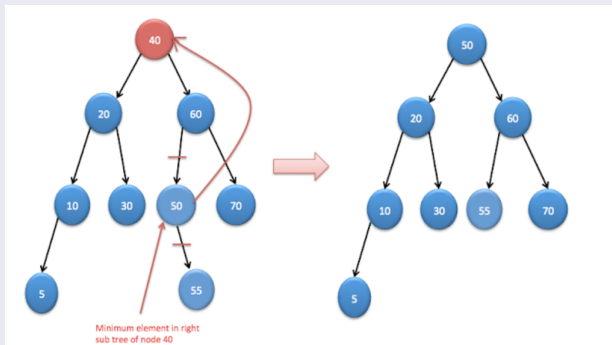
Brisanje čvora sa jednim detetom se svodi na podizanje deteta za jedan nivo na gore (na mestu čvora koji se briše).

Brisanje čvora sa jednim deteta



Brisanje čvora sa jednim detetom se svodi na podizanje deteta za jedan nivo na gore (na mestu čvora koji se briše).

Brisanje čvora sa dva deteta



Brisanje čvora sa dva deteta se svodi na zamenu vrednosti u čvoru koji se briše sa vrednošću u

- krajnjom levom (iz lista) u desnom podstablu čvora koji se briše, ili
- krajnjom desnom u levom podstablu čvora koji se briše, a zatim brisanjem elementa čija vrednost je prebačena

Prezentacija je pripremljena u skladu sa literaturom:

Introduction to Algorithms, 2nd Edition

Autora: Thomas H. Cormen et al.

Napomena: Iz prezentacije ne treba učiti ona nije dovoljna za pripremu ispita, već je dovoljno dobar orijentir koje delove knjige treba da čitate i naučite.