Strukture podataka (120)

Stabla

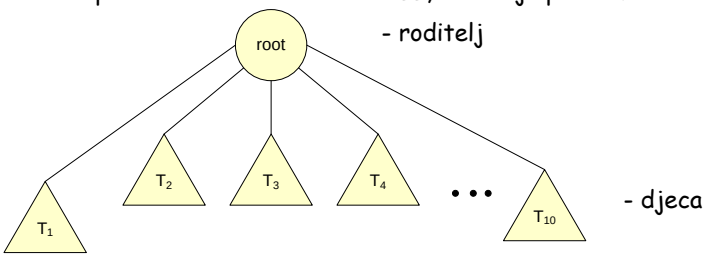
Sadržaj

- Uvod
- Binarna stabla
  - Stablo proračuna
  - Stablo za pretraživanje
  - AVL stabla
  - Kosa stabla
- B stabla

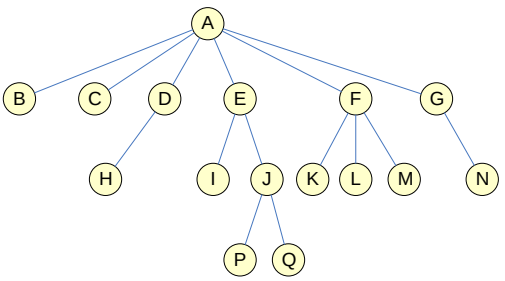
Stabla

- Za velike količine ulaznih podataka linearno vrijeme pristupa (Θ(N)) kod vezanih listi je problem .
- Stabla vrlo korištena apstrakcija, tj. ADT, kod kojeg je u većini slučajeva vrijeme pristupa je Θ(logN).

- Stablo se definira kao skup međusobno povezanih čvorova, pri čemu se početni čvor naziva root.
  - Root čvor sadrži pokazivače na čvorove istog tipa tzv. podstabla.
  - Ukoliko su svi pokazivači u root čvoru NULL, stablo je prazno.



Stabla



- A je root čvor
- čvorovi bez djece su listovi (B, C, H, I, K, L, M, N, P, Q)
- čvorovi s istim roditeljem su braća (npr. K, L i M su potomci od F)

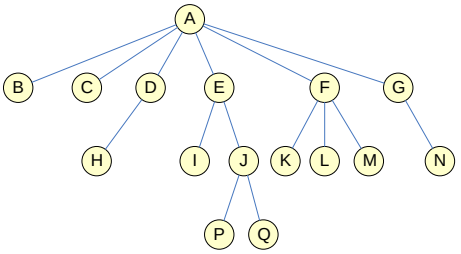
- Do svakog čvora postoji samo jedan put (npr. do P put je: A->E->J->P)

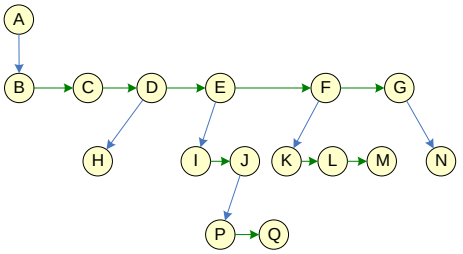
- Dubina (visina) stabla je duljina puta od root čvora do najudaljenijeg lista
  (3 za primjer sa slike).
  - Dubina root čvora je 0.

Stabla

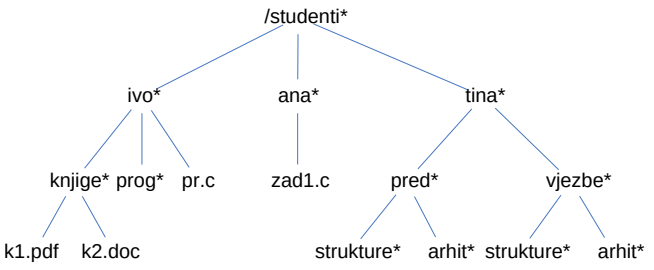
- Implementacija stabla:

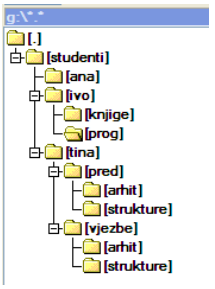
```
typedef struct CvorStabla * Stablo;
struct CvorStabla {
        int El;
        Stablo PrvoDijete;
        Stablo NextBrat;
};
```

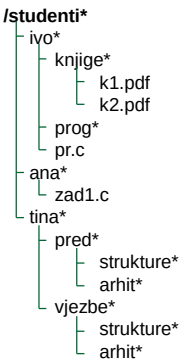




Stabla - primjer

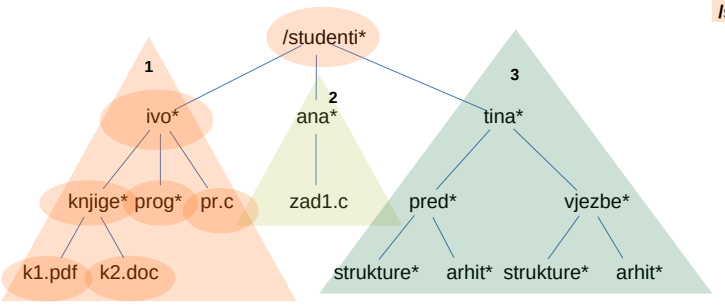






Preorder ispis stabla

- Preorder:
  - prvo se ispiše root čvor
  - zatim se ispisuju njegova djeca s lijeva na desno.
- Ukoliko djeca nisu listovi na njih se primjenjuju ista pravila.

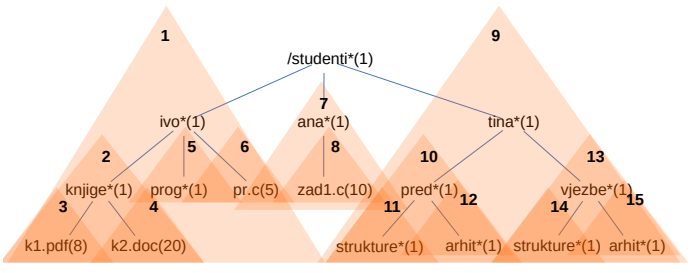



```
/studenti*
  ivo*
    knjige*
      k1.pdf
      k2.pdf
    prog*
    pr.c
  ana*
    zad1.c
  tina*
    pred*
      strukture*
      arhit*
    vjezbe*
      strukture*
      arhit*
```

- Ovaj ispis daje točnu strukturu direktorija

Postorder ispis stabla

- Postorder:
  - prvo se ispišu podstabla djece s lijeva na desno,
  - pa tek onda root čvor.

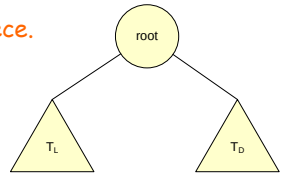



```
        k1.pdf      8
        k2.pdf     20
    knjige*        29
    prog*           1
    pr.c            5
ivo*               36
    zad1.c         10
ana*               11
        strukture*  1
        arhit*      1
    pred*           3
        strukture*  1
        arhit*      1
    vjezbe*         3
tina*               7
/studenti*         54
```

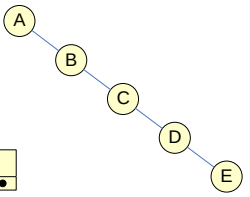
- Korištenjem ovog ispisa može se izračunati veličina direktorija.

Binarna stabla

- Stablo kod kojeg svaki čvor ima najviše dvoje djece.

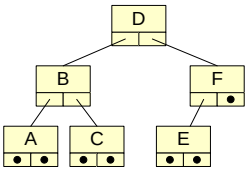


- Najčešće je dubina binarnog stabla znatno manja od broja elemenata tj. dubina je najčešće $\Theta(\sqrt{N})$, dok je za poseban slučaj (binarna stabla pretraživanja) $\Theta(\log(N))$.
- Najgori slučaj binarnog stabla gdje je dubina N-1:



- Implementacija stabla:

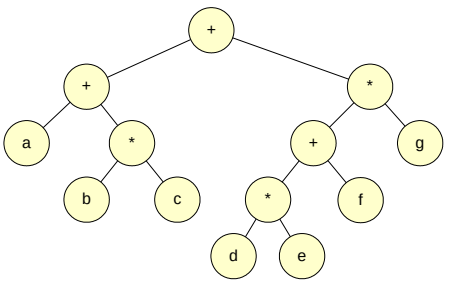
```
typedef struct CvorStabla * Stablo;
struct CvorStabla {
        int El;
        Stablo Lijevo;
        Stablo Desno;
};
```



Stablo proračuna (expression tree)

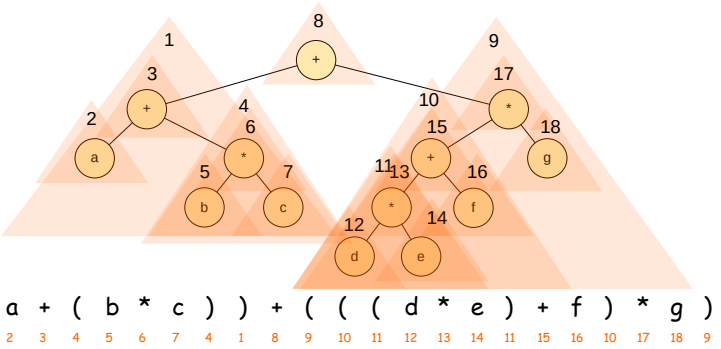
- Jedna od implementacija stabla gdje su operandi listovi, a čvorovi sadrže operacije.
- Ako su operandi binarni, onda je to binarno stablo.

- Za izraz (a+b*c)+((d*e+f)*g) stablo proračuna bi bilo:



Ispis stabla proračuna - inorder

- Stablo proračuna se ispisuje u slijedećem redoslijedu:
  - ispis lijevog podstabla rekurzivno (podstablo se ispisuje u zagradama),
  - ispis operatora,
  - ispis desnog podstabla rekurzivno (podstablo se ispisuje u zagradama).



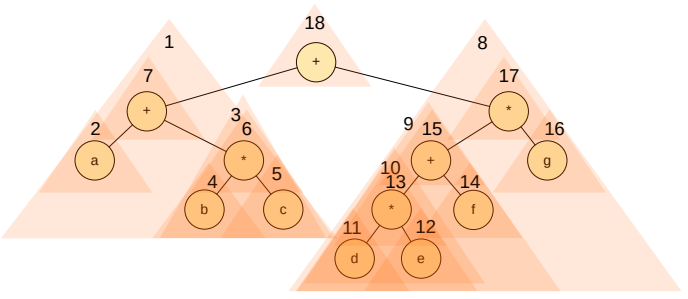
(a + (b * c)) + (((d * e) + f) * g)

1 2 3 4 5 6 7 4 1 8 9 10 11 12 13 14 11 15 16 10 17 18 9

- INORDER prolazak kroz stablo rezultira infix izrazom.

Ispis stabla proračuna - postorder

- Stablo proračuna se ispisuje u slijedećem redoslijedu:
  - ispis lijevog podstabla rekurzivno,
  - ispis desnog podstablo rekurzivno,
  - ispis operatora.



a b c * + d e * f + g * +

2 4 5 6 7 11 12 13 14 15 16 17 18

- POSTORDER ispis rezultira postfix izrazom.

Ispis stabla proračuna - preorder

- Stablo proračuna se ispisuje u slijedećem redoslijedu:
  - ispis operatora,
  - ispis lijevog podstabla rekurzivno,
  - ispis desnog podstabla rekurzivno.

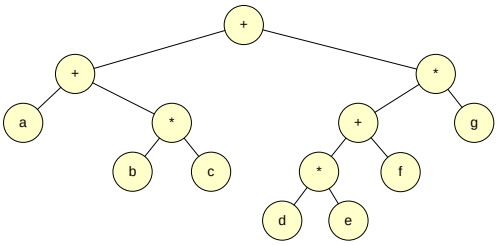
+ + a * b c * + * d e f g

- PREORDER ispis rezultira jednom vrstom postfix izraza koja se manje koristi.

Stvaranje stabla proračuna

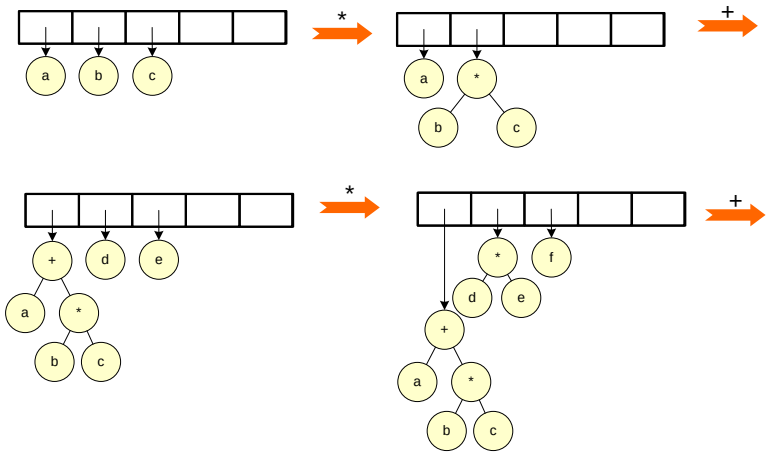
- U stvaranju stabla proračuna kreće se od postfix izraza iz kojeg se čita jedan po jedan simbol, alocira se prostor za taj simbol.
- Ako je simbol:
  - operand, pokazivač na taj čvor se stavlja na stog,
  - operator sa stoga se skidaju dva pokazivača (adrese prethodnih čvorova ili podstabala), prvi postaje desno dijete, a drugi lijevo

- Primjer za: a + b * c + (d * e + f) * g => a b c * + d e * f + g *+



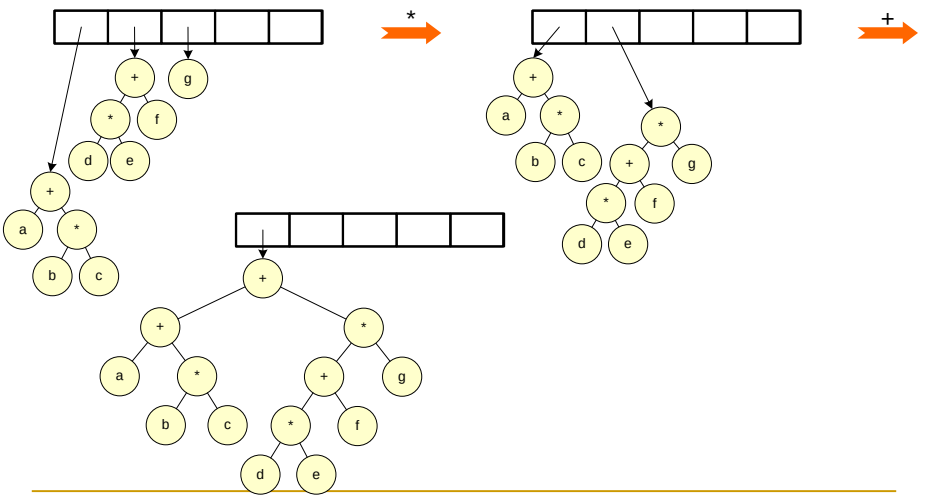
Stvaranje stabla proračuna

a b c * + d e * f + g * +



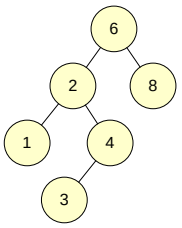
Stvaranje stabla proračuna

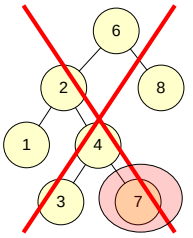
a b c * + d e * f + g * +



Binarna stabla za pretraživanje (binary search tree)

- Jedna od važnih primjena binarnih stabala je za pretraživanje.
- Binarno stablo je **binarno stablo za pretraživanje** ako za svaki čvor X vrijedi:
  - sve vrijednosti u lijevom podstablu su manje od vrijednosti X,
  - sve vrijednosti u desnom podstablu su veće od vrijednosti X.





Binarna stabla za pretraživanje - implementacija

- Zbog rekurzivne definicije stabla, sve funkcije su rekurzivne.

```
struct cvorStabla;
typedef struct cvorStabla Cvor;
typedef struct cvorStabla * Stablo;
typedef struct cvorStabla * Pozicija;

Stablo StvoriPrazno(Stablo S);
Pozicija TraziMin(Stablo S);
Pozicija TraziMax(Stablo S);
Pozicija Trazi(int X, Stablo S);
Stablo Dodaj(int X, Stablo S);
Stablo Brisi(int X, Stablo S);
void Ispis (Stablo S);

struct cvorStabla{
    int El;
    Stablo L;
    Stablo D;
};
```

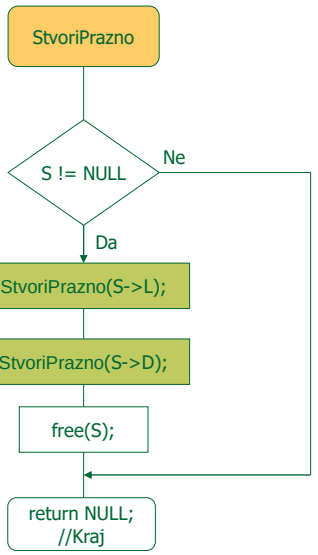
Stablo StvoriPrazno(Stablo S)

- Funkcija StvoriPrazno se koristi uglavnom za inicijalizaciju i ona briše sve elemente stabla ukoliko postoje.
- Kao argument uzima adresu elementa kojeg treba izbrisati ili inicijalizirati.
- Stvori prazno stablo za primjere:

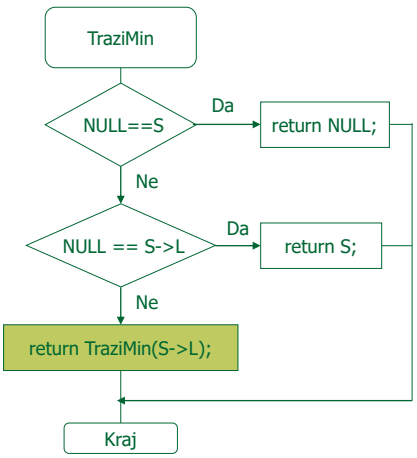
a) root = NULL;

b) root → 6

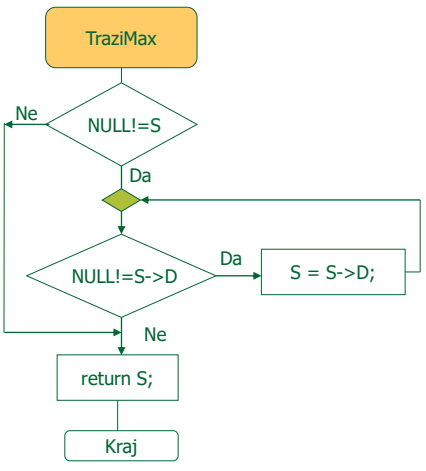
c) root → 6, 4, 7, 2, 5



Pozicija TraziMin (Stablo S)

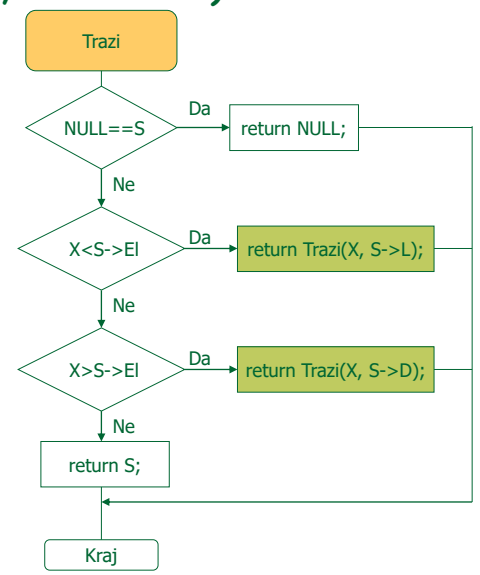


Pozicija TraziMax (Stablo S)



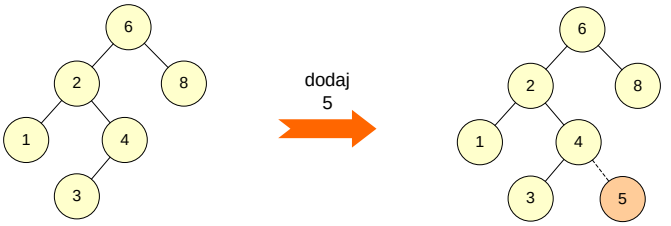
Pozicija Trazi(int X, Stablo S)

- Funkcija Trazi vraća pokazivač na element s vrijednošću X, tj. NULL ako ne postoji.
- Kao argument uzima traženu vrijednost i adresu elementa od kojeg se započinje potraga.

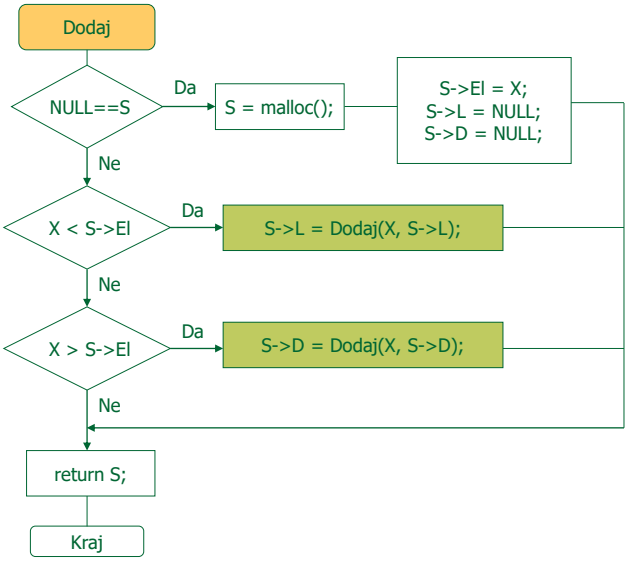


Stablo Dodaj (int X, Stablo S)

- Funkcija Dodaj se koristi za umetanje elementa u stablo i pri tome stablo i dalje treba zadovoljavati uvjete stabla za pretraživanje.

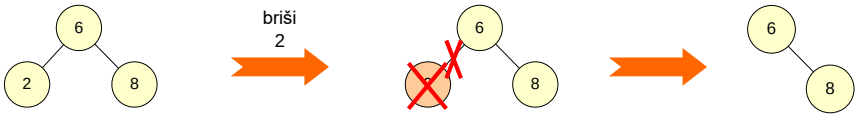


Stablo Dodaj(int X, Stablo S)



Stablo Brisi (int X, Stablo S)

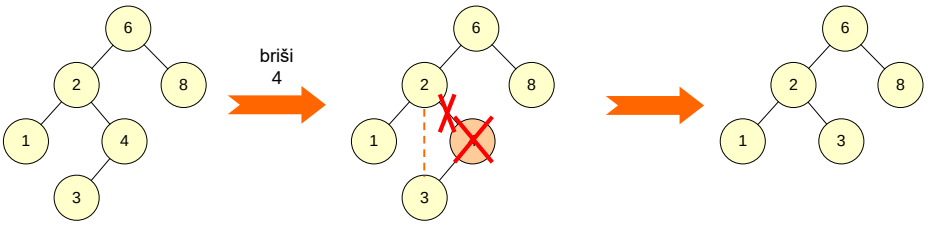
- Brisanje stabla:
  - **brisanje lista**
  - brisanje čvora s jednim djetetom,
  - brisanje čvora s dvoje djece.



1. pronaći čvor
2. izbrisati vezu s roditeljem
3. osloboditi prostor

Stablo Brisi (int X, Stablo S)

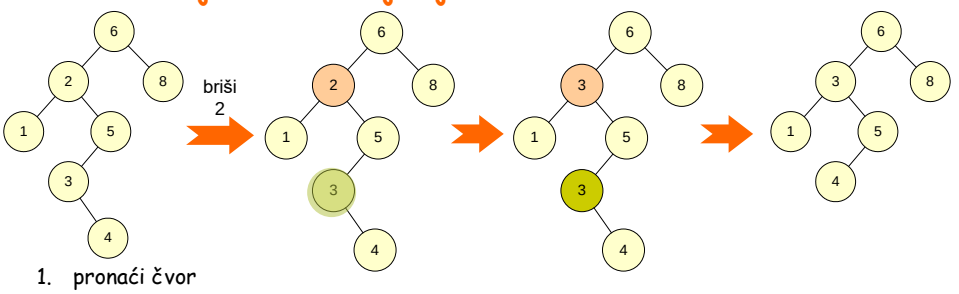
- Brisanje stabla:
  - brisanje lista;
  - **brisanje čvora s jednim djetetom**
  - brisanje čvora s dvoje djece.



1. pronaći čvor
2. uspostaviti vezu između njegovog roditelja i djeteta
3. osloboditi prostor

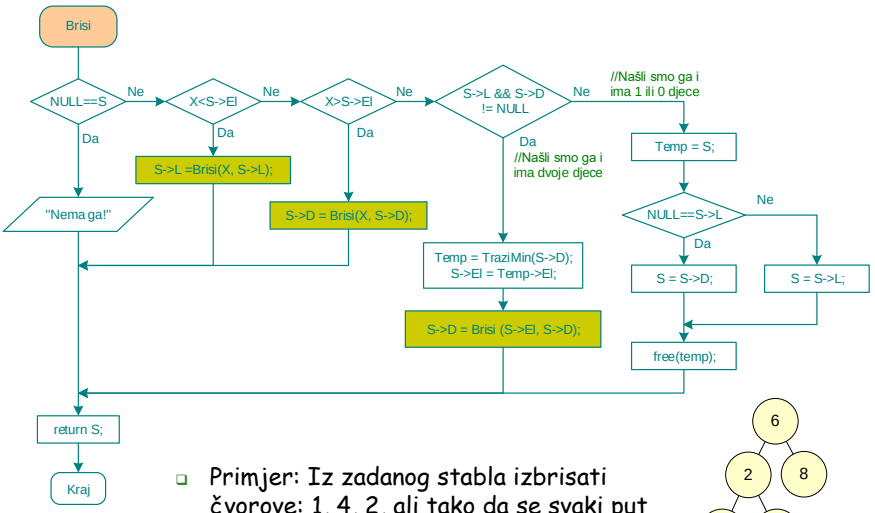
Stablo Brisi (int X, Stablo S)

- Brisanje stabla:
  - brisanje lista;
  - brisanje čvora s jednim djetetom,
  - **brisanje čvora s dvoje djece**



1. pronaći čvor
2. zamijeniti njegovu vrijednost s vrijednošću najmanjeg elementa u desnom podstablu (ili najvećeg u lijevom)
3. izbrisati element koji je sadržavao najmanju vrijednost u desnom podstablu (ili najvećeg u lijevom)

Stablo Brisi (int X, Stablo S)



- Primjer: Iz zadanog stabla izbrisati čvorove: 1, 4, 2, ali tako da se svaki put kreće od početnog stabla.

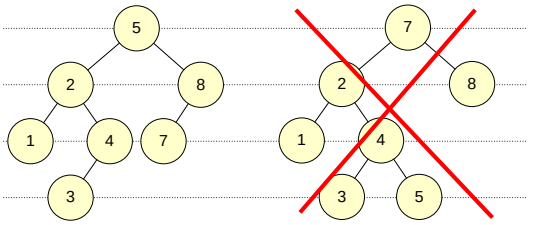
Ispis elemenata binarnog stabla

- Postoji nekoliko da se ispišu elementi stabla:
  - *inorder:*
    - lijevo podstablo, trenutni čvor, desno podstablo;
  - *postorder:*
    - lijevo podstablo, desno podstablo, trenutni čvor;
  - *preorder:*
    - trenutni čvor, lijevo podstablo, desno podstablo;
  - *level order* :
    - svi čvorovi dubine D se obrađuju prije čvorova dubine D+1.

- Kako implementirati funkcije za ove ispise?

AVL stabla – Adelson-Velskii i Landis

- AVL stablo je balansirano binarno stablo pretraživanja.
- Balansirano stablo je stablo kod kojeg za svaki čvor vrijedi da se duljina lijevog i desnog podstabla razlikuju najviše za 1.
- Duljina praznog stabla je –1, a stabla sa samo root čvorom 0.



- Prednost ovog pristupa je u tome što je duljina AVL stabla ≈ logN, pa sve operacije za pretraživanje su Θ(logN)
- Problem je ubacivanje novog člana i brisanje, jer to može uzrokovati balansiranje stabla.

AVL stabla – Umetanje novog elementa

- Kod umetanja novog elementa treba voditi računa da stablo ostane balansirano.
- Kod umetanja promjena balansiranosti se može desiti samo onim čvorovima koji se nalaze između umetnutog i root čvora.
- Promjenu balansiranosti može izazvati:
  - ubacivanje u lijevo podstablo lijevog djeteta,
  - ubacivanje u desno podstablo desnog djeteta,

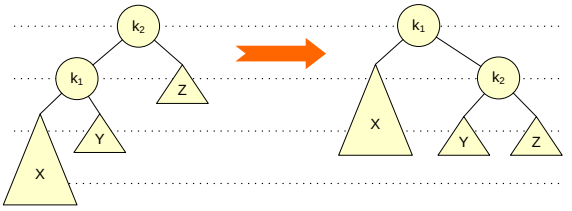
  } jednostruka rotacija

  - ubacivanje u desno podstablo lijevog djeteta,
  - ubacivanje u lijevo podstablo desnog djeteta.

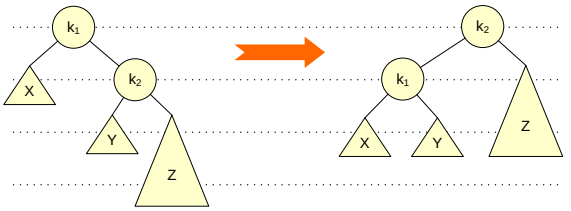
  } dvostruka rotacija

AVL stabla – Jednostruka rotacija

- lijevo podstablo lijevog djeteta

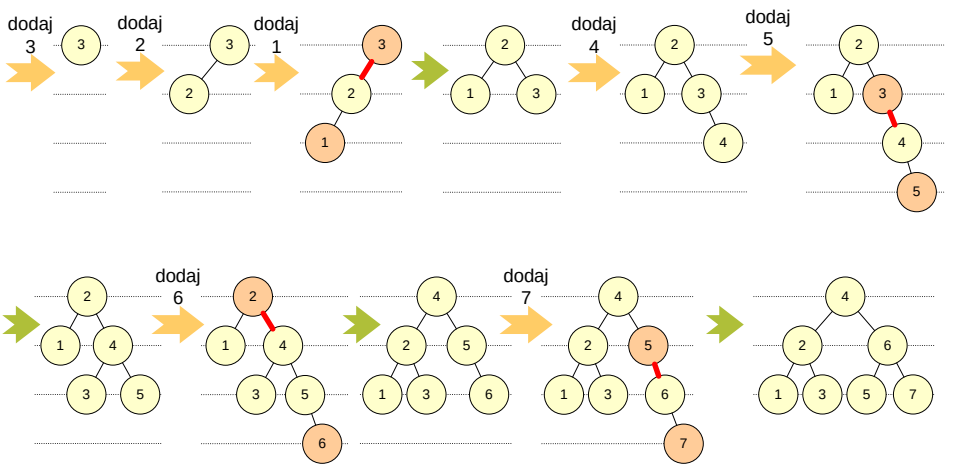


- desno podstablo desnog djeteta



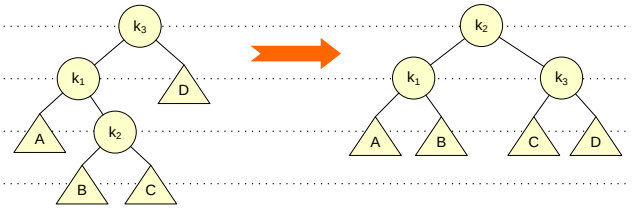
AVL stabla – Jednostruka rotacija

- Zadatak je kreirati AVL stablo a unosi se 3, 2, 1, 4, 5, 6, 7

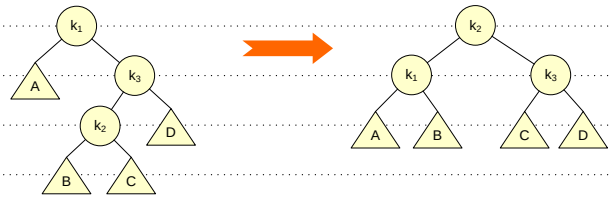


AVL stabla – Dvostruka rotacija

- lijevo-desna dvostruka rotacija

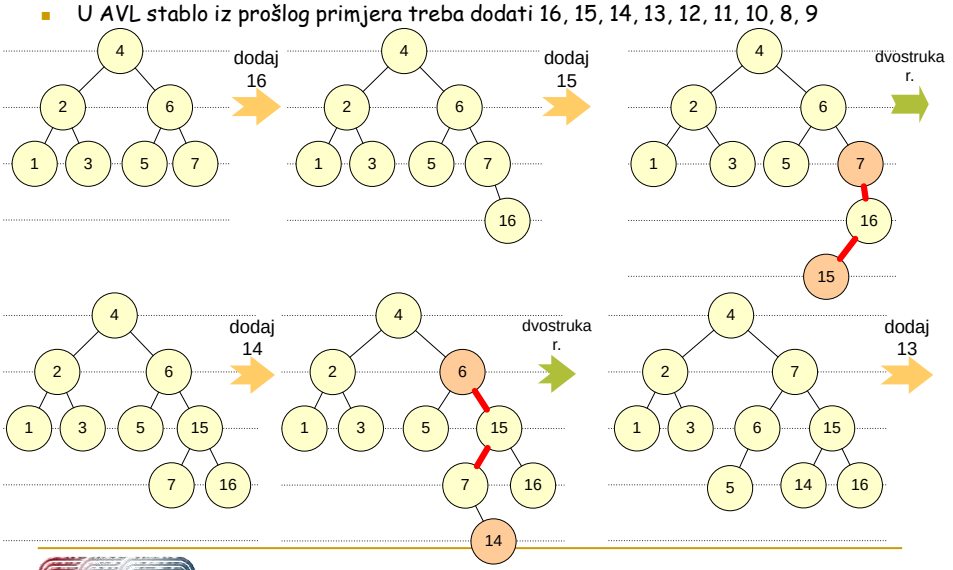


- desno-lijeva dvostruka rotacija

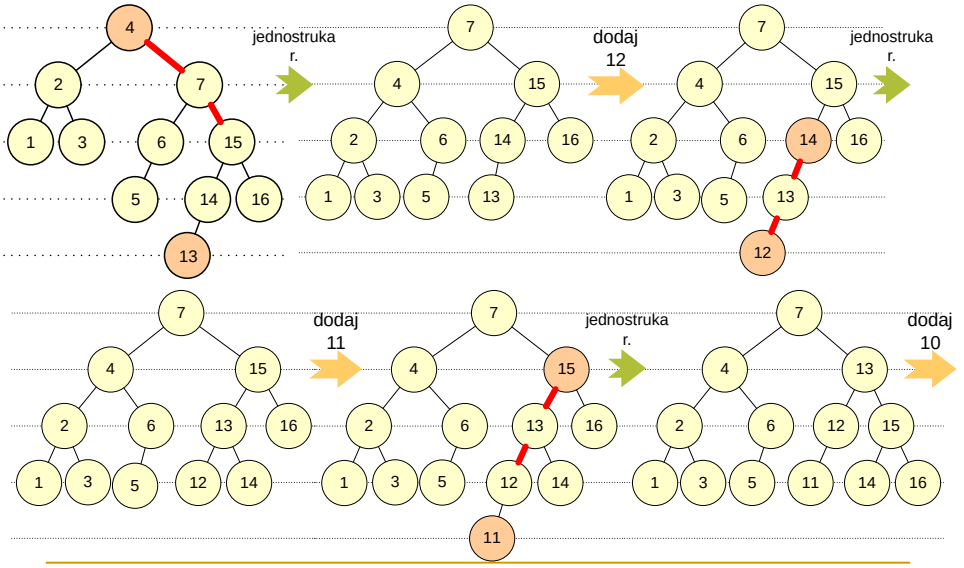


AVL stabla – Dvostruka rotacija

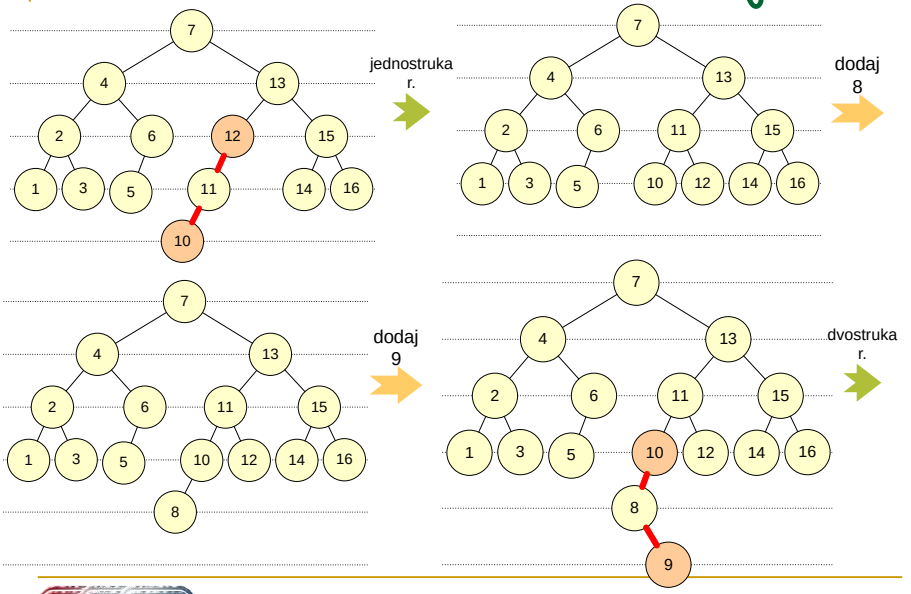
- U AVL stablo iz prošlog primjera treba dodati 16, 15, 14, 13, 12, 11, 10, 8, 9



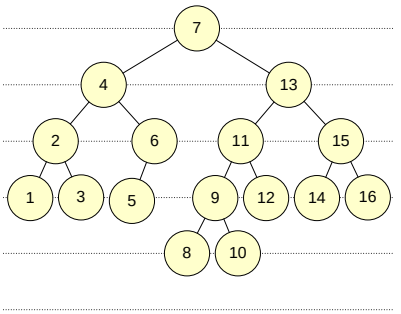
AVL stabla – Dvostruka rotacija



AVL stabla – Dvostruka rotacija



AVL stabla – Dvostruka rotacija

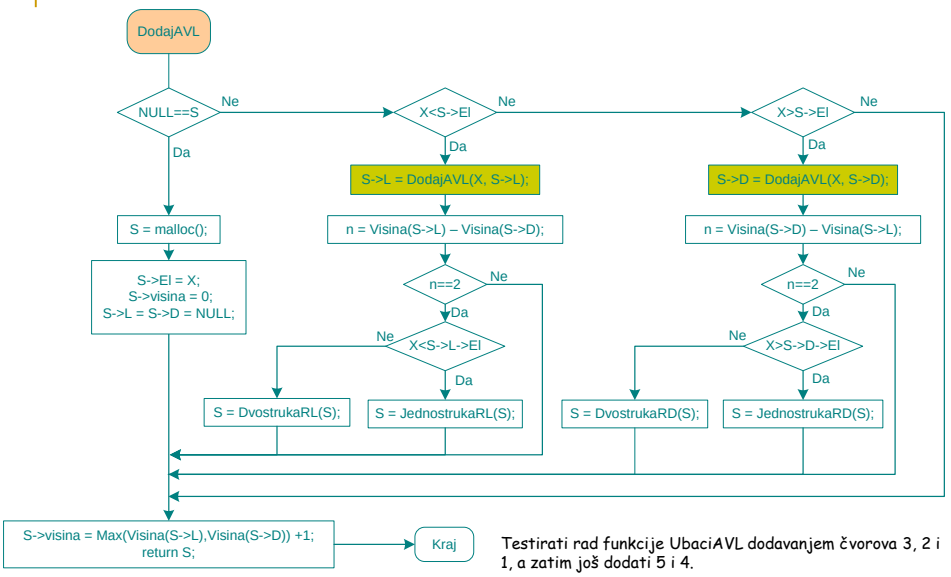


- Napraviti AVL stablo ako se dodaju sljedeći čvorovi: 11 28 35 21 14 7 4 45 48 47 46.

AVL stabla - implementacija

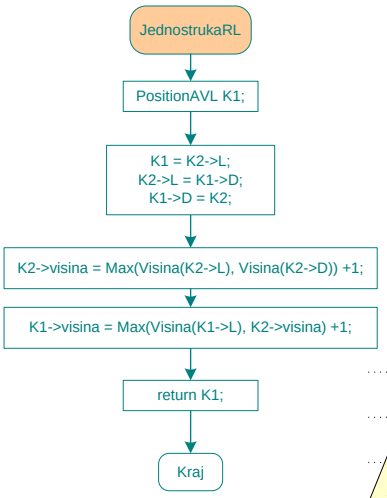
```
struct cvorStablaAVL;
typedef struct cvorStablaAVL CvorAVL;
typedef struct cvorStablaAVL * StabloAVL;
typedef struct cvorStablaAVL * PozicijaAVL;

StabloAVL StvoriPrazno(StabloAVL S);
PozicijaAVL TraziMin(StabloAVL S);
PozicijaAVL TraziMax(StabloAVL S);
PozicijaAVL Trazi(int X, StabloAVL S);
StabloAVL UbaciAVL(int X, StabloAVL S);
StabloAVL Izbrisi(int X, StabloAVL S);

int Visina(StabloAVL S);// -1 ako je S = NULL
PozicijaAVL JednostrukaRL (PozicijaAVL k2);
PozicijaAVL JednostrukaRD (PozicijaAVL k2);
PozicijaAVL DvostrukaRL (PozicijaAVL k3);
PozicijaAVL DvostrukaRD (PozicijaAVL k3);

struct cvorStablaAVL{
    int El;
    StabloAVL L;
    StabloAVL D;
    int visina;
};
```

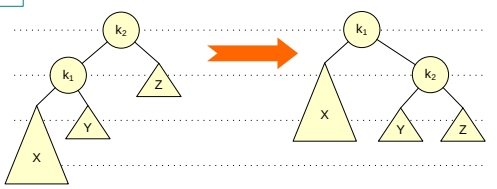
StabloAVL DodajAVL (int X, StabloAVL S)



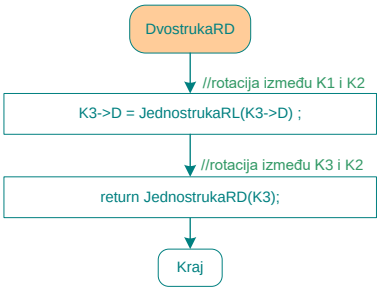
Testirati rad funkcije UbaciAVL dodavanjem čvorova 3, 2 i 1, a zatim još dodati 5 i 4.

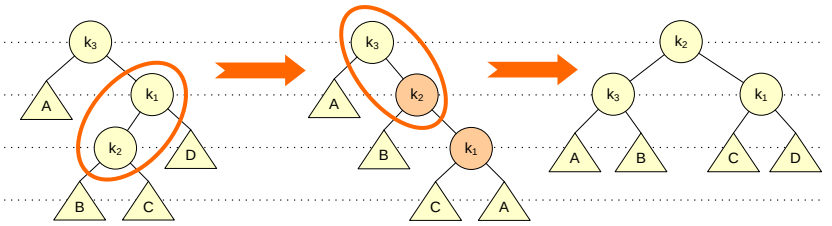
PozicijaAVD JednostrukaRL (PozicijaAVL K2)





PozicijaAVL DvostrukaRD (PozicijaAVL K3)



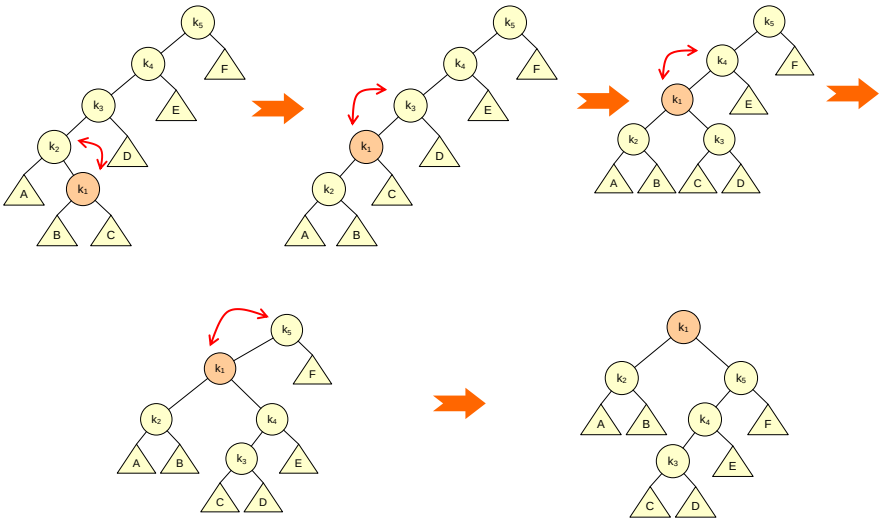


Kosa stabla

- Osnovna ideja kosih stabala je da nakon što se pristupi nekom čvoru, on se pomiče na root poziciju nizom AVL rotacija.
- Ako se taj čvor nalazi duboko u stablu, restrukturiranjem pristup tom i susjednim čvorovima postaje brži.
- Restrukturiranjem se stablo i balansira.
- Unos novog elementa u koso stablo ne traži balansiranje.

- Kosa stabla kreću od pretpostavke da $\Theta(N)$ i nije tako loš, sve dok se događa rijetko.
- Za većinu slučajeva $\Theta(M \log N)$ [M je broj uzastopnih operacija].

Osnovna ideja ukošavanja



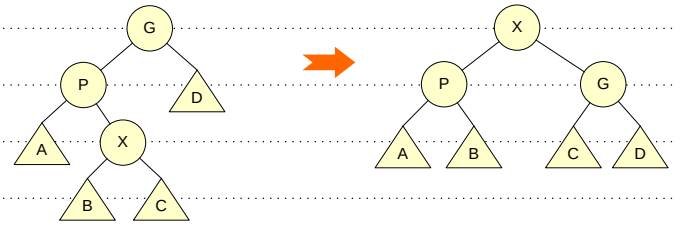
Ukošavanje

- Stvaran pristup ukošavanju je vrlo sličan prikazanoj ideji rotacije, ali se metoda rotacije bira nešto pažljivije.

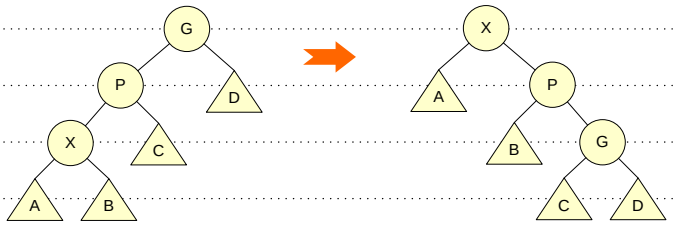
- Ako se uzme da je X čvor kojem se pristupa primjenjuju se slijedeća pravila:
  - ako je roditelj od X root samo se rotira X i root;

  - ako X ima roditelja (P) i pra-roditelja (G) postoje dva rješenja:
    - cik-cak slučaj – X je desno (lijevo), a P lijevo (desno) dijete
    - cik-cik slučaj - X i P su desno (lijevo) dijete.

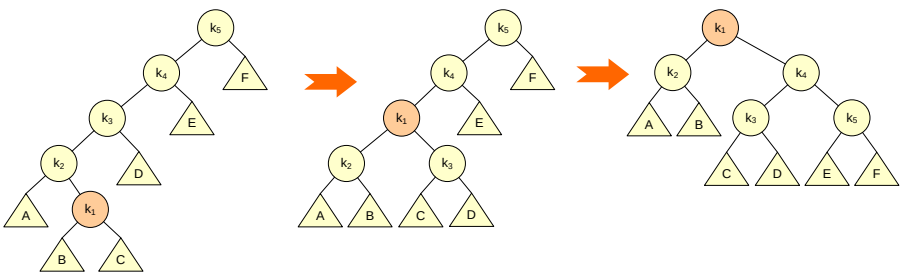
Cik – cak ukošavanje (dvostruka AVL rot.)



Cik – cik ukošavanje

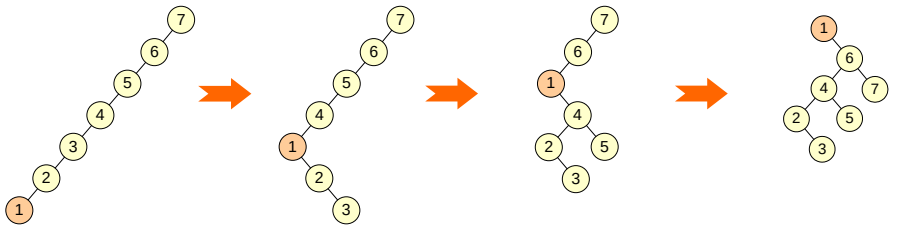


Ukošavanje



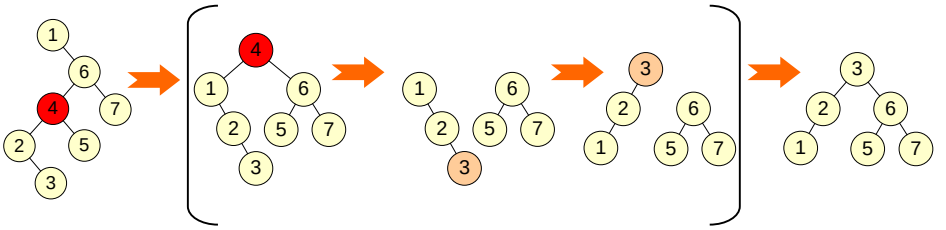
Ukošavanje - primjer

- Kakvim stablom će rezultirati pristup čvoru 1?



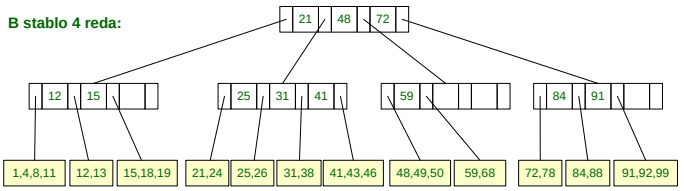
Kosa stabla - brisanje

- Kada se želi izbrisati neki čvor prvo mu se pristupi i time on postaje root čvor.
- Kada se izbriše taj čvor dobiju se dva podstabla T_L i T_R.
- Pristupa se najvećem elementu u T_L i time on postaje root, ali nema desno dijete.
- Za desno dijete se postavi T_R.



B stabla

- Spadaju pod vrlo korištena stabla za pretraživanje koja nisu binarna.

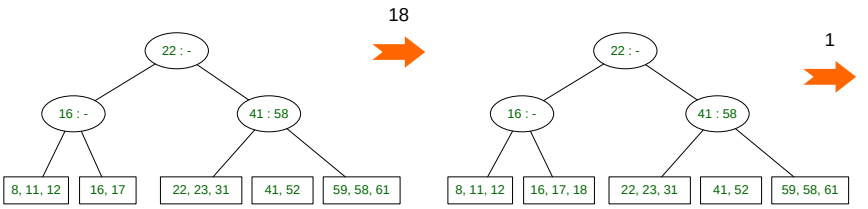


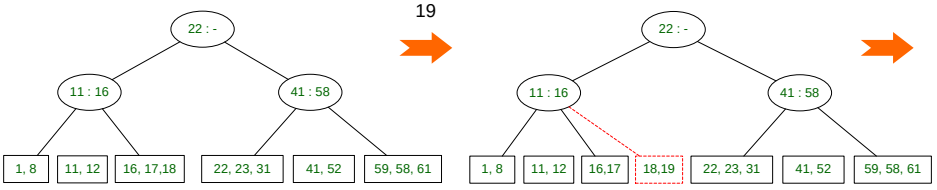
- Za B stablo reda M vrijedi:
  - root čvor je ili list ili ima između 2 i M djece;
  - svi čvorovi, osim roota, imaju između M/2 i M djece;
  - svi su listovi na istoj dubini i sadrže podatke.

- B stablo 4 reda se zove 2-3-4 stablo, dok se stablo 3 reda zove 2-3 stablo.

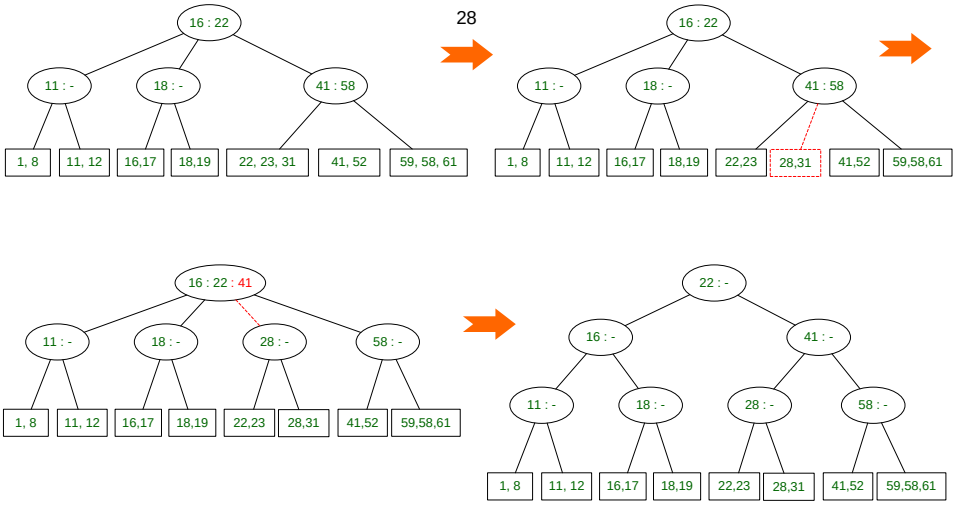
Primjer dodavanja novog elementa u B stablo

- Počevši od zadanog 2-3 stabla ubaciti vrijednosti 18, 1, 19, 28:





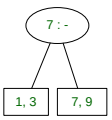
Primjer ubacivanja novog elementa u B stablo



Primjer stvaranja novog B stabla

- Nacrtati 2-3 stablo koje nastaje dodavanjem čvorova 7, 3, 1, 9 u prazno stablo.
  - Ukoliko je stablo 3. reda uzimaju se prva tri elementa i smještaju u čvor

    1, 3, 7
  - Kada stigne četvrti element, za kojeg nema mjesta kreira se novi čvor.



Stabla

Kraj