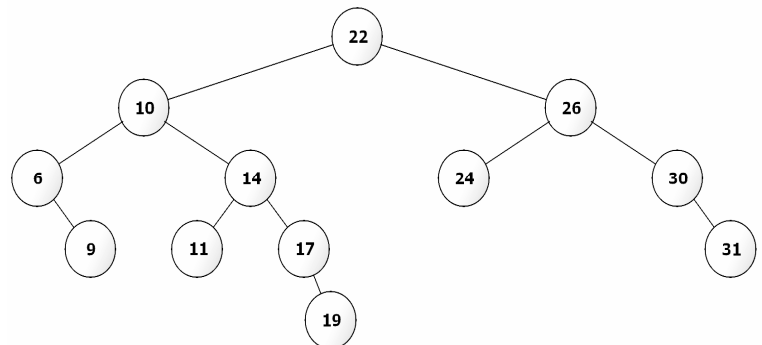


ISPIT IZ ALGORITAMA I STRUKTURA PODATAKA

1. [20] Posmatra se stablo na slici 1.

a) Navesti kriterijum kompletnosti stabla. Da li je stablo kompletno? Da li je stablo balansirano po AVL kriterijumu? U slučaju da stablo nije balansirano po AVL kriterijumu, navesti kritične čvorove.



Slika 1

b) U slučaju da stablo nije AVL stablo, rebalansiranjem ga transformisati u AVL stablo. Potom za dobijeno AVL stablo, prikazati njegov izgled nakon svake od sledećih izmena, redom: umetanje ključeva 29, 34, 18, 20, i 16; brisanje ključeva 14, 22, 26, 29 i 10. **Napomena:** prilikom brisanja, koristiti sledbenika.

2. [30] Predložiti (uz obrazloženje) memorijski efikasnu implementaciju usmerenog retko povezanog težinskog grafa. Usvojiti predloženu implementaciju i na jeziku C (ili C++) napisati potrebne deklaracije, a zatim i funkciju za pronalaženje kritičnog puta u zadatom grafu. Po potrebi napisati dodatne, pomoćne funkcije. Smatrati da je graf formiran pre poziva tražene funkcije. Glavni program ne treba pisati.
3. [25] Dati pseudokod i objasniti opšte algoritme umetanja i pretraživanja ključa u heš tabeli kod tehnika otvorenog adresiranja. Kakav se problem javlja kod brisanja i zašto?
4. [25] Pitanja:
- Izvesti vremensku složenost binarnog pretraživanja. Diskutovati da li ona zavisi od sadržaja niza i zašto?
  - Uporediti algoritme umetanja ključa u B i B+ stablu i objasniti razliku.
  - Objasniti pojam stabilnosti algoritma za sortiranje. Pod kojim uslovima su metodi direktnog umetanja, direktne selekcije i direktne zamene stabilni?