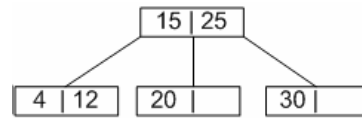


1. [15] **B i B* stabla**

- a) Porediti prema prosečnom broju uspešnih i neuspešnih pristupa **B** i **B*** stablo reda 3 i visine 2 u situaciji kada

- popunjenost svih čvorova je minimalna
- popunjenost svih čvorova je maksimalna

- b) U **B*** stablo prikazano na slici 1 redom se umeću ključevi 22, 17, 13, a zatim se brišu ključevi 4, 12, 25. Prikazati izgled stabla nakon svake izmene.



Slika 1

Izračunati prosečan broj pristupa pri **uspešnoj** pretrazi nakon umetanja svih ključeva i u konačnom stanju.

2. [25] **Uređivanje**

Napisati na jeziku C ili C++ kompletan program koji niz celih brojeva uređuje **nerastuće** primenom stabla selekcije. Postupak uređivanja realizovati u posebnoj funkciji ili metodi. Glavni program treba da pročita niz brojeva sa standardnog ulaza, primeni napisanu funkciju da bi ga uredio i ispiše niz nakon uređivanja.

3. [10] Predložiti izmenu algoritma binarne pretrage za efikasnu pretragu na više ključeva. Diskutovati uređenost niza ključeva na koje se vrši pretraga.

4. [18] Definisati AVL stablo i objasniti svrhu njegovog uvođenja. Objasniti postupke balansiranja u oba karakteristična slučaja uz ilustraciju opštom slikom i uz dokaz korektnosti postupka.

5. [18] Skicirati i objasniti opšte algoritme umetanja i pretraživanja ključa u heš tabeli kod tehnika otvorenog adresiranja. Kakav se problem javlja kod brisanja i zašto?

6. [14]

- a) Navesti algoritme sortiranja kod kojih početni raspored podataka ne utiče na složenost uz objašnjenje zašto je to tako i navesti njihove složenosti
- b) Diskutovati uticaj načina izbora pivota na performanse algoritma *Quicksort* i navesti nekoliko načina.

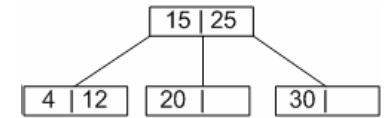
Ispit traje 3 h.

1. [15] **B i B* stabla**

- a) Porediti prema prosečnom broju uspešnih i neuspešnih pristupa **B** i **B*** stablo reda 3 i visine 2 u situaciji kada

- popunjenost svih čvorova je minimalna
- popunjenost svih čvorova je maksimalna

- b) U **B*** stablo prikazano na slici 1 redom se umeću ključevi 22, 17, 13, a zatim se brišu ključevi 4, 12, 25. Prikazati izgled stabla nakon svake izmene.



Slika 1

Izračunati prosečan broj pristupa pri **uspešnoj** pretrazi nakon umetanja svih ključeva i u konačnom stanju.

2. [25] **Uređivanje**

Napisati na jeziku C ili C++ kompletan program koji niz celih brojeva uređuje **nerastuće** primenom stabla selekcije. Postupak uređivanja realizovati u posebnoj funkciji ili metodi. Glavni program treba da pročita niz brojeva sa standardnog ulaza, primeni napisanu funkciju da bi ga uredio i ispiše niz nakon uređivanja.

3. [10] Predložiti izmenu algoritma binarne pretrage za efikasnu pretragu na više ključeva. Diskutovati uređenost niza ključeva na koje se vrši pretraga.

4. [18] Definisati AVL stablo i objasniti svrhu njegovog uvođenja. Objasniti postupke balansiranja u oba karakteristična slučaja uz ilustraciju opštom slikom i uz dokaz korektnosti postupka.

5. [18] Skicirati i objasniti opšte algoritme umetanja i pretraživanja ključa u heš tabeli kod tehnika otvorenog adresiranja. Kakav se problem javlja kod brisanja i zašto?

6. [14]

- a) Navesti algoritme sortiranja kod kojih početni raspored podataka ne utiče na složenost uz objašnjenje zašto je to tako i navesti njihove složenosti
- b) Diskutovati uticaj načina izbora pivota na performanse algoritma *Quicksort* i navesti nekoliko načina.

Ispit traje 3 h.