

Oni koji hoće da im se računa rezultat sa kolokvijuma rade samo zadatke 4, 5 i 6 u trajanju od 2 sata.

Ostali rade sve zadatke u trajanju od 4 sata.

1. [18]Pitanja:

- Izvesti izraz za vezu između interne i eksterne dužine puta u binarnom stablu.
- Objasniti efikasnu memorijsku implementaciju kompletnog i skoro kompletnog binarnog stabla i kretanje kroz takvo stablo.
- Koliki je najmanji, a koliki najveći dozvoljeni broj ključeva u korenu B\*-stabla i zašto?

2. [14] a) Koristeći podatke o prioritetima iz priložene tablice, prikazati po koracima prevođenje sledećeg izraza iz *infix* u *postfix* notaciju:

$$a + (b + c * d) / f \wedge a - ((g + h) \wedge i / j)$$

- Napisati program u simboličkom mašinskom jeziku za 0-adresnu mašinu koji izračunava vrednost izraza iz tačke a), pod pretpostavkom da su sve promenljive smeštene u odgovarajuće memorijske lokacije.

operator	<i>ulpr</i>	<i>spr</i>
-	2	2
*	4	3
+	5	4
/	6	6
$\wedge$	8	7
(	9	0
)	1	-

- [18] Skicirati i objasniti rekurzivni algoritam za generisanje dubinskog obuhvatnog stabla. Izvesti složenost algoritma za obe memorijske reprezentacije grafa. Objasniti kako bi se algoritam realizovao iterativno.
- [20] Napisati na jeziku C ili C++ program za rad sa heš tabelom. Koristi se metoda otvorenog adresiranja. Primarna heš funkcija je  $h_p(k) = k \bmod n$ , gde je  $k$  celobrojni pozitivan ključ, a  $n$  veličina tabele. Kolizije se rešavaju dvostrukim heširanjem sa sekundarnom heš funkcijom  $h_s(k) = m + k \bmod p$ . Vrednosti  $n$ ,  $m$  i  $p$  se zadaju prilikom kreiranja tabele. Realizovati funkciju pretrage tabele na zadati ključ: funkcija vraća vrednost TRUE ako ključ postoji, u suprotnom FALSE. Napisati funkciju za izračunavanje prosečnog broja pristupa pri uspešnoj i neuspešnoj pretrazi. Napisati glavni program koji ilustruje rad sa tabelom.
- [16] Detaljno opisati sortiranje primenom metoda umetanja sa smanjenjem inkrementa. Objasniti na čemu se zasniva efikasnost ovog metoda i kako treba da se bira sekvenca inkremenata. Ilustrovati rad algoritma pri sortiranju niza 21, 72, 55, 47, 12, 88, 81 i 30 u tri iteracije sa efikasnim izborom inkrementa.
- [14] U visinski balansirano AVL stablo umeću se redom ključevi 10, 4, 12, 18, 6, 11, 9, 15, 17, 14, 13 a zatim se redom brišu ključevi 11, 12, 14.
  - Nacrtati izgled stabla nakon svake od navedenih izmena.
  - Izračunati srednji broj pristupa prilikom uspešnog i neuspešnog traženja posle svih umetanja ključeva i u završnom stanju.