

Elektrotehnički fakultet u Beogradu  
Katedra za računarsku tehniku i informatiku

*Predmet:* Algoritmi i strukture podataka (13E112ASP, IR2ASP)

*Nastavnik:* dr Milo Tomašević, red. prof.

*Asistenti:* Marko Mišić, dipl. ing.; Sanja Delčev, dipl. ing.;  
Maja Vukasović, dipl.ing.

*Ispitni rok:* Februar 2017.

*Datum:* 10.02.2017.

*Kandidat\*:* \_\_\_\_\_

*Broj Indeksa\*:* \_\_\_\_\_

*Prvi deo ispita traje 120 minuta. Drugi deo ispita traje 30 minuta.*

*Studenti koji žele da im se prizna domaći zadatak umesto drugog dela ispita treba to da naznače na prvoj stranici.*

*Napuštanje sale nije dozvoljeno tokom prvih 60 minuta.*

*Upotreba literature nije dozvoljena.*

*Zadatak 1* \_\_\_\_\_ /15

*Zadatak 4* \_\_\_\_\_ /10

*Zadatak 2* \_\_\_\_\_ /20

*Zadatak 5* \_\_\_\_\_ /15

*Zadatak 3* \_\_\_\_\_ /10

*Zadatak 6* \_\_\_\_\_ /15

**Prvi deo ispita:** \_\_\_\_\_/80

**Drugi deo ispita:** \_\_\_\_\_/20

**Ukupno na ispitu:** \_\_\_\_\_/100

**Napomena:** Ukoliko u postavci nekog zadatka postoje nepreciznosti, student treba da uvede razumnu pretpostavku, da je u okviru (da bi se lakše prepoznala prilikom ocenjivanja) i da nastavi da izgrađuje preostali deo svog odgovora na temeljima uvedene pretpostavke. Kod pitanja koja imaju ponuđene odgovore treba **samo zaokružiti** jedan odgovor. Na ostala pitanja odgovarati **čitko, kratko i precizno**.

\* popunjava student.

---

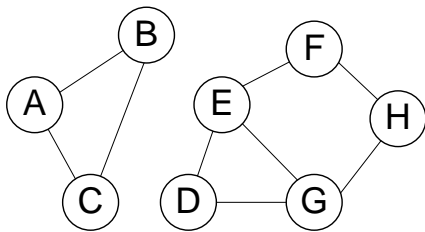
## Prvi deo ispita (strane 2 - 6)

---

### 1. [15] Određivanje puta u grafu

- a) [8] Predložiti efikasan način za određivanje matrice puta u neusmerenom netežinskom grafu. Komentarisati složenost postupka ukoliko se koriste matrice susednosti.

- b) [7] Za graf sa slike, prikazati po koracima rad predloženog algoritma i odrediti konačnu matricu puta.



2. [20] Neka se skupovi celih brojeva predstavljaju pomoću stabla binarnog pretraživanja.

a) [5] Objasniti na koji način se najefikasnije može odrediti unija dva takva skupa.

b) [15] Napisati u pseudokodu implementaciju funkcije koja određuje uniju dva skupa i formira treći skup, kao rezultat. Obratiti pažnju da visina rezultujućeg stabla bude što manja. Dozvoljena je upotreba rekurzije i dodatnih struktura podataka.

BST SET UNION(*root1*, *root2*)

3. [10] Prikazati izgled zadatog niza prilikom sekvencijalne pretrage na sekvencu ključeva 4, 11, 5, 4, 13, 7, 13, 4, ukoliko se koristi metoda prebacivanja na početak.

|    |    |   |   |    |   |   |    |   |   |
|----|----|---|---|----|---|---|----|---|---|
| 15 | 13 | 3 | 4 | 14 | 7 | 9 | 11 | 1 | 5 |
|    |    |   |   |    |   |   |    |   |   |
|    |    |   |   |    |   |   |    |   |   |
|    |    |   |   |    |   |   |    |   |   |
|    |    |   |   |    |   |   |    |   |   |
|    |    |   |   |    |   |   |    |   |   |
|    |    |   |   |    |   |   |    |   |   |
|    |    |   |   |    |   |   |    |   |   |
|    |    |   |   |    |   |   |    |   |   |

4. [10] Data je sekvenca ključeva 25, 26, 25, 30, 25, 26, 27, 25, 25, 25, 30, 25, 26. Predložiti algoritam za sortiranje tako da ono bude vremenski što efikasnije. Diskutovati izbor konkretnog algoritma i prikazati postupak sortiranja date sekvence ključeva po koracima.

5. [10] Za koje grafove se može koristiti *Dijkstra*-in algoritam? Kolika je složenost za obe predstave grafa? U nekoliko rečenica objasniti kako se do nje dolazi.

6. [15] Napisati i objasniti pseudokod funkcije koja briše zadati ključ  $K$  iz heš tabele. Heš funkcija je  $h(K) = K \bmod n$ , za razrešavanje kolizije se koristi linearno pretraživanje, a pri brisanju se radi linearno pomeranje.

DELETE( $H, K$ )

---

## Drugi deo ispita – programski zadatak (strane 7 - 8)

---

1. [20] Formirano je B+ stablo reda 3 umetanjem nekog skupa ključeva koji su celi brojevi. Funkciji `getAncestor` prosleđuju se donja i gornja granica nekog intervala, kao i pokazivač na koren stabla. Implementirati ovu funkciju na jeziku C ili C++ tako da vraća adresu korena najmanjeg podstabla čiji svi ključevi pripadaju datom intervalu. Ako takav ne postoji, funkcija treba da vrati `null`. Struktura svih čvorova u stablu je ista i ne postoji pokazivač na pretke.

```
Node* getAncestor(Node* root, int low, int high)
```

