

---

Elektrotehnički fakultet u Beogradu  
Katedra za računarsku tehniku i informatiku

*Predmet:* Algoritmi i strukture podataka 2 (13S112ASP2)

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

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

*Ispitni rok:* Treći kolokvijum (januar 2018. godine)

*Datum:* 21.01.2018.

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

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

*Kolokvijum traje 100 minuta. Napuštanje sale nije dozvoljeno tokom prvih 60 minuta.  
Upotreba literature nije dozvoljena.*

<i>Zadatak 1</i>	_____ /5	<i>Zadatak 5</i>	_____ /10
<i>Zadatak 2</i>	_____ /20	<i>Zadatak 6</i>	_____ /15
<i>Zadatak 3</i>	_____ /15	<i>Zadatak 7</i>	_____ /10
<i>Zadatak 4</i>	_____ /15	<i>Zadatak 8</i>	_____ /10

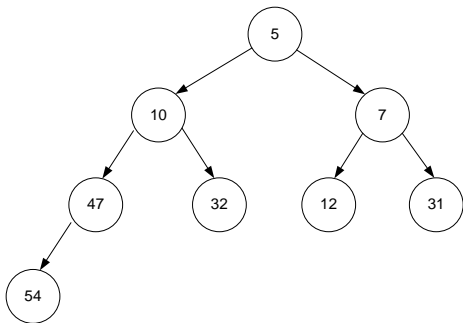
**Ukupno na kolokvijumu:** \_\_\_\_\_ /100

**Napomena:** Ukoliko u postavci nekog zadatka postoje nepreciznosti, student treba da uvede razumnu pretpostavku, da je uokviri (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.

---

1. [5] Prikazati izgled **binarnog hipa** nakon uklanjanja minimalnog ključa i nakon umetanja ključeva 9 i 30.



2. [20] Napisati iterativnu implementaciju sledećeg algoritma heširanja. Za heširanje se koriste dve heš funkcije,  $h_1(K)$  i  $h_2(K)$ , dok se za razrešavanje kolizija koristi sekundarna heš funkcija  $g(k)$ . Nad ključem  $K$  koji se ubacuje u tabelu se primenjuju i funkcija  $h_1(K)$  i  $h_2(K)$ , a kao ulaz se koristi rezultat funkcije za koju se dobije manje kolizija. U slučaju jednakog broja kolizija, koristi se funkcija  $h_1(K)$ . Potrebno je realizovati operacije umetanja i pretrage.

INSERT(table, K)

SEARCH(table, K)

3. [15] Data je sekvenca ključeva: 169, 84, 99, 6, 61, 84, 9, 96, 158, 42, 495, 46, 223. Prikazati rad *shell sort* algoritma po koracima. Navesti korišćenu sekвенцу inkremenata i obrazložiti izbor.

4. [15] Priloženi pseudokod prikazuje jednu varijantu *bubble sort* algoritma za sortiranje podataka. Objasniti glavne nedostatke priloženog algoritma i napisati u pseudokodu alternativnu implementaciju ovog algoritma koja te nedostatke ispravlja.

```
for i:=1 to n-1 do
  for j:=n-1 downto 1 do
    if arr[j]>arr[j+1] then
      b:=arr[j]; arr[j]:=arr[j+1]; arr[j+1]:=b;
    end_if
  end_for
end_for
```

BUBBLE SORT REVISITED(*arr, n*)

5. [10] Podaci se smeštaju u heš tabelu sa 7 ulaza primenom heš funkcije  $h_p(K) = K \bmod 7$ . Za razrešavanje kolizija se koristi metoda dvostrukog heširanja sa sekundarnom heš funkcijom  $h_s(K) = 4 + K \bmod 3$ . Odrediti prosečan broj pristupa prilikom neuspešnog traženja i verovatnoću popunjavanja praznih ulaza, pod pretpostavkom da su svi ključevi jednako verovatni.

0	14
1	
2	11
3	24
4	18
5	
6	13

6. [15] Napisati u pseudokodu funkciju za uklanjanje minimalnog ključa u **binomnom hipu** i njegovu naknadnu reorganizaciju. Nije potrebno sprovesti potencijalno spajanje stabala nakon uklanjanja čvora sa minimalnim ključem. Binomni hip predstavljen pomoću objekta klase *BinHeap* koja sadrži ulančanu listu objekata *BinTree* koji predstavljaju pojedinačna binomna stabla. Lista je uređena rastuće po stepenu stabla. Binomno stablo se sastoji od čvorova predstavljenih objektima klase *BinNode*. Funkcija kao argument prima objekat klase *BinHeap*. U nastavku je dat spisak metoda koje su na raspolaganju, njihove povratne vrednosti i klase objekata za koje ih treba pozvati.

Metoda	Povratna vrednost	Objekat klase
<i>getFirst</i>	pokazivač na početak liste šume binomnih stabala	<i>BinHeap</i>
<i>findMin</i>	pokazivač na binomno stablo u čijem korenu je minimalni ključ	<i>BinHeap</i>
<i>getOrder</i>	red stabla	<i>BinTree</i>
<i>setOrder</i>	nema	<i>BinTree</i>
<i>getRoot</i>	pokazivač na koren binomnog stabla	<i>BinTree</i>
<i>getNumOfChildren</i>	broj potomaka čvora binomnog stabla	<i>BinNode</i>
<i>getChildAtPos(index)</i>	pokazivač na čvor koji je potomak objekta <i>BinNode</i> na poziciji <i>index</i>	<i>BinNode</i>

REMOVE MIN KEY(*binheap*)

7. **[10]** Podaci se smeštaju u heš tabelu sa 8 ulaza primenom heš funkcije po metodu deljenja. Za razrešavanje kolizija se koristi metoda slučajnog pretraživanja ( $c = 3$ ). Prikazati proces razrešavanja kolizije i generisati ispitne nizove za ključeve 21 i 18. Objasniti da li kod ovog metoda u opštem slučaju dolazi do primarnog i sekundarnog grupisanja i zašto.
8. **[10]** Precizno objasniti kako se dolazi do složenosti algoritama za sortiranje a) kvadratnom selekcijom b) stablom selekcije i c) spajanjem. U kojem od ovih algoritama se razlikuju najbolji, srednji i najgori slučaj složenosti.