

---

Elektrotehnički fakultet u Beogradu  
Katedra za računarsku tehniku i informatiku

*Predmet:* Algoritmi i strukture podataka 2 (13S112ASP2)  
*Nastavnici:* dr Milo Tomašević, red. prof.; doc. dr Marko Mišić  
*Asistenti:* Sanja Delčev, dipl. ing.; Maja Vukasović, dipl.ing.;  
Balša Knežević, dipl.ing.  
*Ispitni rok:* Prvi kolokvijum (novembar 2021.)  
*Datum:* 03.11.2021.

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

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

*Kolokvijum traje 120 minuta, prvih sat vremena nije dozvoljeno napuštanje sale.  
Upotreba literature nije dozvoljena.*

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

**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. [15] Neki programski kod je preveden u mašinski kod. Radi jednostavnije analize mašinskog koda, koristi se alat za prikaz generisanih instrukcija. Programski kod je podeljen u funkcije, a svaka funkcija predstavlja određenu programsku celinu koja se prevodi u skup instrukcija. Nakon prevođenja, tela funkcija, odnosno instrukcije koje ih čine su smeštene susedno u memoriji. Često korišćena funkcionalnost u alatu je pretraga određene instrukcije prema njenoj adresi u memoriji. Međutim, takav upit nije moguće izvršiti direktno. Umesto toga, za svaku funkciju postoji informacija o adresi prve instrukcije koja pripada toj funkciji. Nakon selektovanja funkcije kojoj pripada tražena instrukcija, otvara se prozor sa svim instrukcijama koje joj pripadaju čime se omogućava pretraga i dohvaćanje te instrukcije.
- a) [7] Opisati koja struktura podataka se koristi za čuvanje informacija o adresama instrukcija programa. Kako funkcioniše pretraga?

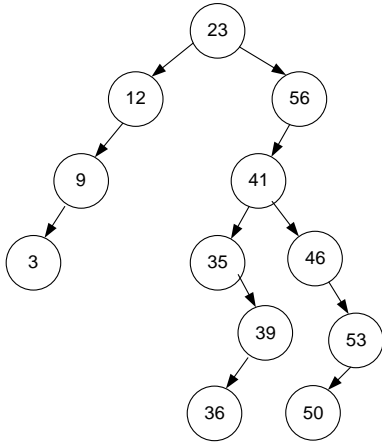
- b) [8] Neka je program podeljen u tri funkcije tako da svaka sadrži instrukcije na adresama navedenim u tabeli. Prikazati sadržaj strukture podataka opisane pod a) u koju su smeštene ove operacije i prikazati postupak pretrage instrukcije na adresi 25.

Func0	Func1	Func2
4	15	23
7	18	25
11	21	28
13		31

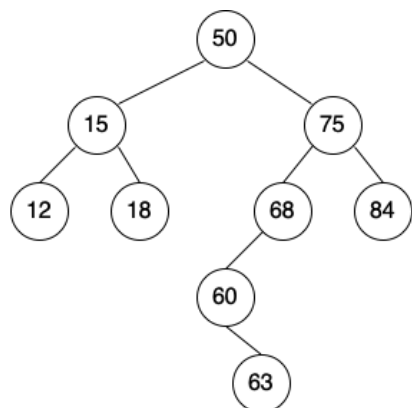
2. [10] Dat je niz koji predstavlja cenu koka-kole u  $n$  različitih prodavnica. Niz cena je **uređen** neopadajuće. Perica kod sebe ima  $x$  para i želi da kupi koka-kolu. Napisati u pseudokodu **efikasan** algoritam koji izračunava za koji broj prodavnica Perica **nema** dovoljno novca da kupi koka-kolu. Parametar *price* predstavlja niz cena,  $n$  predstavlja dužinu niza, dok  $x$  predstavlja količinu novca kojom raspolaže Perica.

CALC SHOP NUM(*price*,  $n$ ,  $x$ )

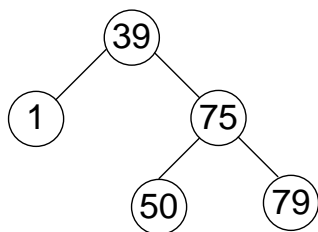
3. [10] Na slici je dato stablo binarnog pretraživanja. Potrebno je transformisati ovo stablo u AVL stablo. Prikazati transformaciju stabla po koracima.



4. [10] Na slici je prikazano stablo binarnog pretraživanja. Iz stabla se redom brišu ključevi: 12, 15, 50, 60, 63, 68, 75, 84, 18. Prikazati izgled stabla nakon svakog brisanja. Pri brisanju koristiti sledbenika.



5. [10] Neka se u dato samopodešavajuće stablo redom vrše sledeće operacije: umetanje ključeva 31, 37 i 4, brisanje ključa 37, umetanje ključa 43, pretraga ključa 26 i brisanje ključa 44. Prikazati izgled stabla nakon svakog izvršenog koraka.



6. [15] Dat je *postorder* obilazak binarnog stabla pretraživanja, nizom *post* dužine *n*. Napisati iterativnu efikasnu funkciju u pseudokodu kojom se na osnovu datog obilaska rekonstruiše stablo. Funkcija treba da vrati koren rekonstruisanog stabla.

BST-RECONSTRUCT(*post, n*)

7. [15] Neka se posmatra jedno AVL stablo. Ukoliko je dat pokazivač na čvor *node* koji je upravo umetnut u stablo, napisati u pseudokodu implementaciju funkcije koja ažurira balanse čvorova i pronalazi najbliži (najdublji) kritičan čvor u odnosu na mesto umetanja. Ukoliko takav ne postoji, funkcija treba da vrati *NULL* vrednost. Smatrati da čvor poseduje pokazivač na oca, kao i informaciju o balansu čvora.

AVL FIND CRITICAL NODE(*node*)

8. [15] Suboptimalna stabla binarnog pretraživanja.

- a) [7] Neka je dat skup od  $n$  ključeva sa poznatim verovatnoćama uspešnog pretraživanja  $p_i$ , dok su verovatnoće neuspešnog pretraživanja  $q_i$  jednake 0. Ukoliko se formira suboptimalno stablo binarnog pretraživanja samo na osnovu verovatnoća uspešnog pretraživanja  $p_i$ , dati predlog heuristike koja to omogućava i navesti odgovarajući, specijalizovani izraz za računanje cene tako dobijenog stabla.

- b) [8] Neka se suboptimalno stablo binarnog pretraživanja tehnikom predloženom pod a) formira od skupa ključeva 10, 18, 24, 33, 46, 51, 55, 66 sa poznatim verovatnoćama uspešnog pretraživanja datim u tabeli, formirati i nacrtati takvo suboptimalno stablo i izračunati njegovu cenu. Obrazložiti odgovor.

$K_i$	<b>10</b>	<b>18</b>	<b>24</b>	<b>33</b>	<b>46</b>	<b>51</b>	<b>55</b>	<b>66</b>
$p_i$	0,15	0,02	0,12	0,03	0,3	0,25	0,08	0,05