

## Binarna stabla

1. Napisati funkciju *broj\_cvorova* koja izračunava broj čvorova u datom binarnom stablu.
2. Napisati funkciju *visina* koja izračunava visinu datog binarnog stable.
3. Napisati funkciju *identicna* koji provjerava da li su dva data binarna stabla identična.
4. Napisati funkciju *listovi* koji štampa sve listove datog binarnog stabla.
5. Napisati program koji za dato binarno stablo, čiji su elementi cijeli brojevi štampa list na najmanjoj dubini. Ako ima više listova na najmanjoj dubini, treba odštampati onaj sa najmanjom vrijednošću.
6. Napisati program koji za dato binarno stablo ispisuje elemente na najvećoj dubini.
7. Napisati program koji štampa sve čvorove binarnog stabla koji imaju dva sina.
8. Napisati funkciju *prebroj* koja vraća broj čvorova binarnog stabla koji su iste parnosti kao oba njihova sina. Ukoliko je čvor list ili ima samo jednog sina, ne uzimati ga u obzir.
9. Napisati funkciju *suma1* koja vraća sumu svih čvorova binarnog stabla koji su veći od sume svojih sinova. Čvorove koji nemaju potomke ne uzimati u obzir.
10. Napisati funkciju *suma2* koja sumu svih parnih elemenata binarnog stabla na dubini  $d$  umanjuje za sumu svih neparnih elemenata na dubini  $d$ .
11. Napisati funkciju *simetricno* kojom se provjerava da li je binarno stablo simetrično. Binarno stablo je simetrično ukoliko je ako povučemo vertikalnu liniju kroz korijen, desno podstablo odraz u ogledalu lijevog podstabla.
12. Napisati funkciju *balansirano* koja provjerava da li je dato binarno stablo visinski izbalansirano, tj. da li za svaki čvor stabla važi da se visina njegovog lijevog i desnog podstabla razlikuju najviše za 1.
13. Napisati program koji provjerava da li dato binarno stablo predstavlja validno binarno stablo pretraživanja.
14. Napisati program koji štampa čvorove binarnog stabla po nivoima. Voditi računa o vremenskoj složenosti algoritma. (Hint: koristiti određenu strukturu podataka za obilazak čvorova).
15. Napisati funkciju *sirina* koja ima jedan parametar korijen, koji predstavlja pokazivač na korijen binarnog stabla, a vraća širinu stabla (maksimalan broj čvorova na jednom nivou stabla). Pokušati implementaciju funkcije koja tačno jednom prolazi svakim čvorom.
16. Napisati funkciju *krajnji\_cvorovi* koja ima jedan parametar korijen, koji predstavlja pokazivač na korijen binarnog stabla. Funkcija treba da štampa krajnji lijevi i krajnji desni čvor na svakom nivou stabla. Pokušati implementaciju funkcije koja tačno jednom prolazi svakim čvorom.
17. Napisati program koji učitava prirodan broj  $A$  sa standardnog ulaza, a zatim kreira binarno stablo na sljedeći način. Čvor sa vrijednošću  $A$  postaje korijen stabla. Dalje se za svaki čvor stabla sa vrijednošću  $X$  računaju  $X_1 = X/4$  i  $X_2 = X - 16$ . Ako je  $X_1 > 0$ , čvor  $X$  dobija lijevog sina sa vrijednošću  $X_1$ . Ako je  $X_2 > 0$ , čvor  $X$  dobija desnog sina sa vrijednošću  $X_2$ .  
Za maksimalan broj poena, stablo je potrebno kreirati u jednom prolazu (bez vraćanja u korijen stabla prilikom dodavanja svakog novog čvora)

18. Napisati funkciju *najduza\_putanja* koja štampa najdužu putanju od korijena binarnog stabla do najudaljenijeg lista.
19. Napisati funkciju *suma\_na\_putanji* koja provjerava da li u binarnom stablu u kojem se čuvaju cijeli brojevi postoji putanja od korijena do nekog lista takva da je suma čvorova na toj putanji jednaka datom broju s.
20. Implementirati AVL stablo (<https://www.geeksforgeeks.org/introduction-to-avl-tree/>)
  - a. Dodavanje - <https://www.geeksforgeeks.org/insertion-in-an-avl-tree/>
  - b. Brisanje - <https://www.geeksforgeeks.org/deletion-in-an-avl-tree/>