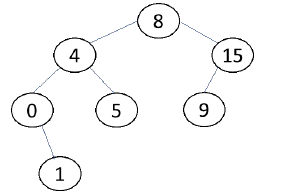
**algoritmi i strukture  
podataka**

**20.09.2019.**

**Zadatak 1.**

Zadano je binarno stablo:



Ako koristimo preorder način obilaska stabla i pri tome ispisujemo vrijednosti pohranjene u čvorovima stabla, dobit ćemo ispis:

|  |
| --- |
| 8, 4, 0, 1, 5, 15, 9 |
| 1, 0, 5, 4, 9, 15, 8 |
| 0, 1, 4, 5, 8, 9, 15 |
| 15, 9, 8, 5, 4, 1, 0 |
| 8, 4, 15, 0, 5, 9, 1 |

**Objašnjenje**

Postoje tri načina obilaska stabla:

1. preorder: korijen, lijevo dijete, desno dijete
2. inorder: lijevo dijete, korijen, desno dijete
3. postorder: lijevo dijete, desno dijete, korijen

Ako dano stablo obilazimo preorder načinom, posjećujemo redom 8, 4, 0, 1, 5, 15, 9.

**20.09.2019.**

**Zadatak 2.**

Što će se ispisati izvođenjem sljedećeg programa?

#include <stdio.h>

void f2(int n)  
{  
 if (n >= 2) return;  
 f2(n + 1);  
 printf("%d ", n);  
}

void f1(int n)  
{  
 printf("%d ", n);  
 if (n >= 2) return;  
 f2(n + 1);  
}

int main(void)  
{  
 f1(-1);  
 return 0;  
}

|  |
| --- |
| -1 |
| -1 2 1 0 |
| -1 1 0 |
| -1 1 0 -1 |
| -1 0 1 |

**Objašnjenje**

Program započinje rad pozivom funkcije f1 uz argument -1 koja ispisuje vrijednost argumenta i, u slučaju da broj nije veći ili jednak dva, poziva funkciju f2.

void f1(int n) // n = -1  
{  
 printf("%d ", n); // ispis -1  
 if (n >= 2) return; // uvjet nije zadovoljen  
 f2(n + 1); // poziva se f2(0)  
}

Funkcija f1 poziva rekurzivnu funkciju f2 s parametrom 0.

Prvo pokretanje funkcije f2:

void f2(int n) // n = 0  
{  
 if (n >= 2) return; // uvjet nije zadovoljen  
 f2(n + 1); // poziva se f2(1)  
 printf("%d ", n); // ispis 0 nakon povratka iz rekurzije  
}

Drugo pokretanje funkcije f2 (rekurzija):

void f2(int n) // n = 1  
{  
 if (n >= 2) return; // uvjet nije zadovoljen  
 f2(n + 1); // poziva se f2(2)  
 printf("%d ", n); // ispis 1 nakon povratka iz rekurzije  
}

Treće pokretanje funkcije f2 (rekurzija):

void f2(int n) // n = 2  
{  
 if (n >= 2) return; // izlaz iz funkcije  
 f2(n + 1);   
 printf("%d ", n);  
}

Konačno, ispis programa u pravilnom vremenskom redoslijedu je -1 1 0.

**21.09.2018.**

**Zadatak 1.**

Neka polje sadrži n cijelih brojeva. Koliko je vrijeme sortiranja polja *quicksort* algoritmom u najboljem slučaju i koliko je vrijeme sortiranja polja *insertion sort* algoritmom u najboljem slučaju?

|  |
| --- |
| quicksort: , insertion sort: |
| quicksort: , insertion sort: |
| quicksort: , insertion sort: |
| quicksort: , insertion sort: |
| quicksort: , insertion sort: |

**Objašnjenje**

Objašnjeno u prezentacijama iz predmeta.