



## Predaja zadaće

Zadaća se predaje do utorka 21.3.2023 u 22.00h. Vašu zadaću predajte u folder zadaca3 na vaš github repozitorij tako da teorijski dio nazovete teorijaPrezime.pdf, a programerski dio nazovite BSTPrezime.cpp.

---

**Zadatak 1.** Za skup ključeva  $\{13, 19, 43, 63, 66, 87, 92\}$ . Nacrtajte binarna stabla pretraživanja (u nastavku BST) visina 2, 3, 4, 5, 6 (samo krajnji rezultat).

---

**Zadatak 2.** Neka je  $T$  BST u kojem su svi ključevi različiti. Dokažite da ako čvor  $x$  nema desno dijete i  $y$  je sljedbenik od  $x$  onda je  $y$  najniži predak od  $x$  čije dijete je isto predak od  $x$ .

---

**Zadatak 3.** Neka je  $T$  BST u kojem su svi ključevi različiti, neka je  $x$  list i  $y$  roditelj od  $x$ . Dokažite da vrijedi točno jedna od sljedeće dvije tvrdnje:

1.  $y$ .ključ je najmanji ključ u  $T$  veći od  $x$ .ključ
  2.  $y$ .ključ je najveći ključ u  $T$  manji od  $x$ .ključ
- 

**Zadatak 4.** Dokažite da neovisno na kojem čvoru započeli u BST-u visine  $h$ , za  $k$  uzastopnih poziva TREE-SUCCESSOR potrebno je  $O(k + h)$  vremena.

---

**Zadatak 5.** Alternativna metoda za inorder ispis BST-a od  $n$  čvorova pronalazi minimalni element u stablu pozivajući TREE-MINIMUM, a zatim radi  $n - 1$  TREE-SUCCESSOR poziva. Dokažite da ovaj algoritam radi u  $O(n)$  vremenu.

---

**Zadatak 6.** Pretpostavimo da konstruiramo BST tako da ubacujemo različite ključeve uzastopno. Argumentirajte da je broj ispitanih čvorova u pretraživanju za neki ključ  $1 +$  broj čvorova ispitanih za prvo ubacivanje tog ključa u BST.

---

**Zadatak 7.** Možete sortirati skup od  $n$  ključeva tako da prvo izgradite BST koji sadrži sve te ključeve (koristeći TREE-INSERT za ubacivanje svakog redom), a onda ispišete ključeve inorder ispisom. Koje je worst-case i best-case vrijeme izvršavanja ovako danog algoritma, svoje tvrdnje potkrijepite analizom pseudokoda TREE-INSERT-a.

---



**Zadatak 8.** Je li operacija brisanja iz BST-a komutativna, u smislu da ako prvo obrišemo čvor  $x$ , a onda čvor  $y$  i dobijemo stablo onda je to stablo jednako stablu koje se dobije prvo brisanjem čvora  $y$ , a onda brisanjem čvora  $x$ . Agrumentirajte, ako nije dajte kontraprimjer.

---

**Zadatak 9.** Implementirajte strukturu BST-a (samo za pozitivne cijele brojeve) koristeći sljedeće upute:

1. Pred vama je dan predlozak koda po kojem trebate napraviti svoj BST:

```
#include <iostream>

using namespace std;

struct Node
{
    int key;
    Node *left, *right, *parent;
};

class BST
{
    Node *root;

    Node* insert(int key, Node *node)
    {
    }

    void inorder(Node *node)
    {
    }

    Node* search(int key, Node *node)
    {
    }

    Node* find_min(Node *node)
    {
    }

    Node* find_max(Node *node)
    {
    }
}
```



```
    }

    Node* successor(Node *node)
    {

    }

    Node* predecessor(Node *node)
    {

    }

    Node* BSTdelete(int key, Node *node)
    {

    }

public:
    BST() : root(NULL) {}

    void print()
    {

    }

    void insert(int key)
    {

    }

    bool search(int key)
    {

    }

    int find_min()
    {

    }

    int find_max()
    {

    }
```



```
int successor(int key)
{

}

int predecessor(int key)
{

}

void BSTdelete(int key)
{

}

};

int main()
{

}
```

2. U main dijelu koda napravite sljedeće:

- Izradite BST i u njemu ubacite ključeve 5, 10, 4, 8, 9, 1, 11, 15, 3 redom.
- Ispišite sve čvorove u stablu u sortiranom poretku.
- Pronađite minimalni i maksimalni element stabla te ih ispišite.
- Pronađite sljedbenika čvora s ključem 8 i prethodnika čvora s ključem 10 te ih oba ispišite.
- Izbrišite element s ključem 10 iz stabla, a zatim još jednom ispišite sve čvorove u stablu u sortiranom poretku.

