Lab 05: Binary Search Tree

Data Structure 2023

Binary Search Tree

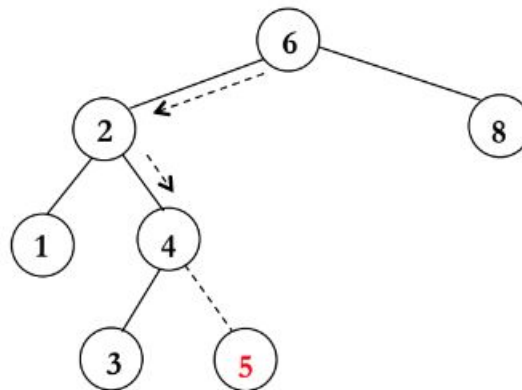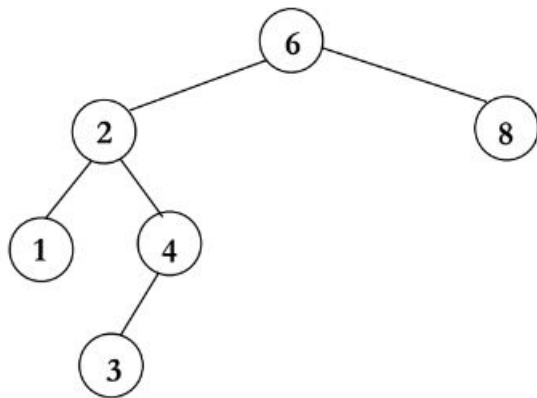- **Which one can be the Binary Search Tree?**
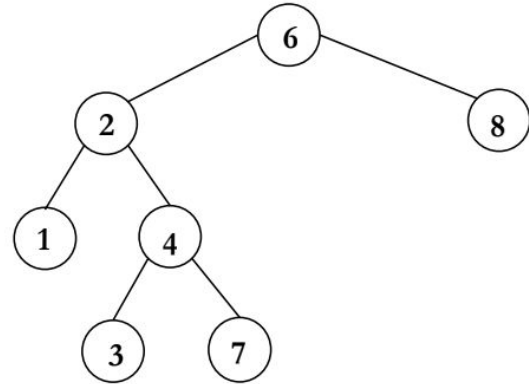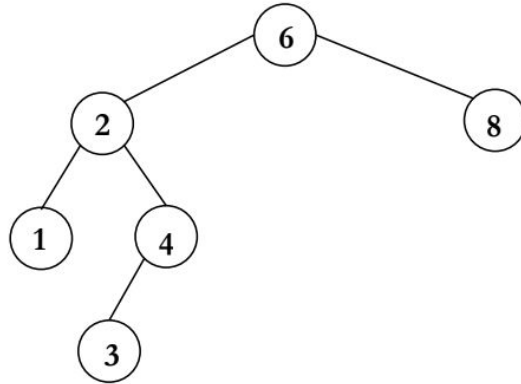
Binary Search Tree

- **For every node X in the tree,**
    - The values of all the keys in its left subtree are smaller than the key value in X.

    - The values of all the keys in its right subtree are larger than the key value in X.

HANYANG UNIVERSITY

Binary Search Tree ADT

- **insertNode** Insert a new node with the key value into the tree. If the key already exists in the tree, print an error message.

- **deleteNode** Delete a node with the given key value from the tree. Reform tree using right subtree. If the key does not exist in the tree, print an error message.

- **findNode** Find the key value in the binary search tree. If the key does not exist, print an error message.

- **printInorder** Print the tree by inorder traversal.

- **deleteTree** Delete the tree.

HANYANG UNIVERSITY
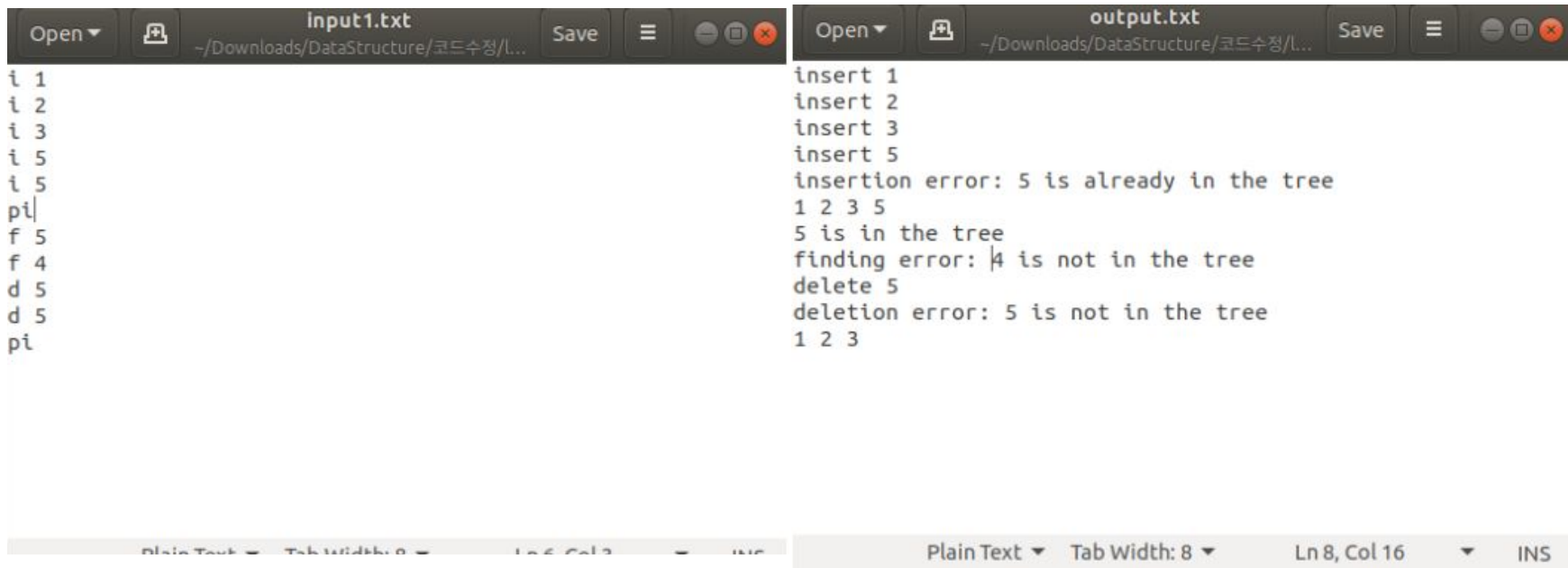
Binary Search Tree ADT

Structure

```
Typedef struct BST* Tree;

typedef struct BST{

    int value;

    struct BST* left;

    struct BST* right;

}BST;
```

Function

```
Tree insertNode(Tree root, int key);

Tree deleteNode(Tree root, int key);

int findNode(Tree root, int key);

void printInorder(Tree root);

void deleteTree(Tree root);
```

Input & Output Example

input1.txt (~/Downloads/DataStructure/코드수정/l...)
```
i 1
i 2
i 3
i 5
i 5
pi
f 5
f 4
d 5
d 5
pi
```

output.txt (~/Downloads/DataStructure/코드수정/l...)
```
insert 1
insert 2
insert 3
insert 5
insertion error: 5 is already in the tree
1 2 3 5
5 is in the tree
finding error: 4 is not in the tree
delete 5
deletion error: 5 is not in the tree
1 2 3
```

HANYANG UNIVERSITY

Assignment

- Due
  - **~ 2022.04.12(수) 23:59**
  - Last Commit 기준

- 자세한 내용은 과제 명세 PDF 파일 참고

HANYANG UNIVERSITY