

9

DATA STRUCTURES

Lab Exercise # 9: Binary Search Tree

In this lab exercise, you will implement a Binary Search Tree data structure. The starter code includes the **BST** class and your task is to complete the constructor, destructor and the following missing methods:

Note: You are NOT allowed to add more methods or change the signature of the methods in the starter code.

1. **void insert(Node* ptr, int key, string value)** (2 Points)

This method inserts a new Node with a given key and value (passed as arguments to the method) in a tree/subtree rooted at ptr.

2. **void remove(Node* ptr, int key)** (1 Points)

This method removes a node with a specific key (passed as an argument to the method) from the tree/subtree which is rooted at ptr.

3. **Node* getRoot()** (1 Points)

This method returns a pointer to the root of the Tree.

4. **int height(Node* ptr)** (1 Points)

This method finds and returns the height (maximum depth) of a tree/subtree rooted at ptr.

5. **Node* find(Node* ptr, int key)** (1 Points)

This method finds and returns a pointer to the node with the specific key (passed as an argument to the method) in a tree/subtree rooted at ptr. The method should return nullptr if the key is not found.

6. **Node* findMin(Node* ptr)** (1 Points)

This method finds and returns a pointer to the Node with the minimum key in a tree/subtree rooted at ptr.

7. **Node* findMax(Node* ptr)** (1 Points)

This method finds and returns a pointer to the Node with the maximum key in a tree/subtree rooted at ptr.

8. **void printInorder(Node* ptr)** (1 Points)

This method traverses the tree in-order and prints the keys of the tree/subtree which is rooted at ptr.

Comments: (1 Points)

Comments are a very important part of any program. You should always write comments in your code, even if not explicitly asked.

Desired Output:

```
> g++ lab11.cpp && ./a.out
```

display	:Display the BST Tree
height	:Find the hieght of the Tree
min	:Find the node with minimum key in BST
max	:Find the node with maximum key in BST
find <key>	:Find a node with a given key value in BST
insert <key,value>	:Insert a new node in BST
remove <key>	:Remove the node from BST
inorder	:Print the BST in Inorder
help	:Display the available commands
exit	:Exit the program

```
>demo
```

```
Tree:
```

```
      10
     /  \
    5    13
   / \  / \
  4  7 12 15
```

```
>height
```

```
Height = 2
```

```
>min
```

```
Min. key = 4
```

```
>max
```

```
Max. key = 15
```

```
>remove 7
```

```
Tree:
```

```
      10
     /  \
    5    13
   / \  / \
  4   12 15
```

```
>remove 5
```

```
Tree:
```

```
      10
     /  \
    4    13
   / \  / \
      12 15
```

```
>remove 10
```

```
Tree:
```

```
      12
     /  \
    4    13
           \
            15
```

```
>insert 5,v5
```

```
Tree:
```

```
    12
```

```
    4
```

```
    13
```

```
    5
```

```
    15
```

```
>inorder
```

```
(4,v4) (5,v5) (12,v12) (13,v13) (15,v15)
```

```
>find 13
```

```
(13,v13)
```

```
>find 10
```

```
10 not found
```

```
>exit
```

CODE OF CONDUCT

All assignments are graded, meaning we expect you to adhere to the academic integrity standards of NYU Abu Dhabi. To avoid any confusion regarding this, we will briefly state what is and isn't allowed when working on an assignment/lab-task.

Any documents and program code that you submit must be fully written by yourself. You can, of course, discuss your ideas with fellow students, as long as these discussions are restricted to general solution techniques. Put differently, these discussions should not be about concrete code you are writing, nor about specific results you wish to submit. When discussing an assignment with others, this should never lead to you possessing the complete or partial solution of others, regardless of whether the solution is in paper or digital form, and independent of who made the solution, meaning you are also not allowed to possess solutions by someone from a different year or course, by someone from another university, or code from the Internet, etc. This also implies that there is never a valid reason to share your code with fellow students, and that there is no valid reason to publish your code online in any form.

Every student is responsible for the work they submit. If there is any doubt during the grading about whether a student created the assignment themselves (e.g. if the solution matches that of others), we reserve the option to let the student explain why this is the case. In case doubts remain, or we decide to directly escalate the issue, the suspected violations will be reported to the academic administration according to the policies of NYU Abu Dhabi.

<https://students.nyud.nyu.edu/campus-life/community-standards/policies/academic-integrity/>