January 2023 CSE 106
Online Assignment on BST
Height of the tree and Depth of a node

Time: 30 minutes

Subsections C1 & C2

- **Depth:** of a node refers to the number of its ancestors, excluding the node itself. So the root node is at depth 0.
- **Height:** of a tree refers to the maximum depth of any node in the tree + 1

In this online, you have to determine the height of the BST. This operation is to be done in $\mathcal{O}(1)$ time. You also have to find the depth of a given key. This should take $\mathcal{O}(\log n)$ time.

Input

Two new commands,

- Depth (P) followed by an integer denoting the value of the node for which the depth is to be calculated Returns: the depth of the node if it exists. -1 otherwise
- Height (H) Returns: The height of the BST at its current stage.

Output

Please output in a file.
Exact match with the output is expected for a faster evaluation.

Mark distribution

Task 1 (30%) Implement depth function in $\mathcal{O}(\log n)$.

Task 2 (70%) Implement height function in $\mathcal{O}(1)$. Solving it in $\mathcal{O}(n)$ will bear only 20% of the total mark.

Sample I/O

Input

```
F 1
I 8
I 10
I 3
P 3
H
I 1
I 14
I 6
I 4
I 13
I 7
P 4
P 6
P 7
```

```
P 8
P 14
P 15
H
T In
T Pre
T Post
D 8
D 7
D 10
D 10
P 14
P 10
P 13
D 4
H
F 4
```

Output

```
not found
(8)
(8(,10))
(8(3,10))
1
2
(8(3(1,),10))
(8(3(1,),10(,14)))
(8(3(1,6),10(,14)))
(8(3(1,6(4,)),10(,14)))
(8(3(1,6(4,)),10(,14(13,))))
(8(3(1,6(4,7)),10(,14(13,))))
3
2
3
0
2
-1
4
1 3 4 6 7 8 10 13 14
8 3 1 6 4 7 10 14 13
1 4 7 6 3 13 14 10 8
(10(3(1,6(4,7)),14(13,)))
(10(3(1,6(4,)),14(13,)))
(13(3(1,6(4,)),14))
(13(3(1,6(4,)),14))
1
-1
0
(13(3(1,6),14))
3
not found
```

Input

```
I 1
H
I 2
H
I 3
```

```
H
I 4
H
I 5
H
I 6
H
I 7
H
I 9
H
I 10
H
I 8
H
D 1
H
D 5
H
D 2
H
D 3
H
D 4
H
D 9
H
D 8
H
D 10
H
D 6
H
D 7
H
```

Output

```
(1)
1
(1(,2))
2
(1(,2(,3)))
3
(1(,2(,3(,4))))
4
(1(,2(,3(,4(,5)))))
5
(1(,2(,3(,4(,5(,6))))))
6
(1(,2(,3(,4(,5(,6(,7)))))))
7
(1(,2(,3(,4(,5(,6(,7(,9)))))))))
8
(1(,2(,3(,4(,5(,6(,7(,9(,10)))))))))
9
(1(,2(,3(,4(,5(,6(,7(,9(8,10)))))))))
9
(2(,3(,4(,5(,6(,7(,9(8,10)))))))))
8
(2(,3(,4(,6(,7(,9(8,10)))))))
```

```
7
(3(,4(,6(,7(,9(8,10))))))
6
(4(,6(,7(,9(8,10)))))
5
(6(,7(,9(8,10))))
4
(6(,7(,10(8,))))
4
(6(,7(,10)))
3
(6(,7))
2
(7)
1
()
0
```

Hints

1. Don't you think the depth and search are very similar?
2. For the height of the tree, you have to add a height variable to every node. Then properly maintain nodes' height during the insertion and deletion operation.
3. Recursion formula for height = 1 + max(height of the left bst, height of the right bst)

Submission Guideline

1. Create a directory with your 7 digit student id as its name
2. Put the source files only into the directory created in step 1
3. Zip the directory (compress in .zip format; .rar, .7z or any other format is not acceptable)
4. Upload the .zip file on moodle.

For example, if your student id is 215xxx, create a directory named 2105xxx. Put only your source files (.c, .cpp, .java, .h, etc.) into 215xxx. Compress 215xxx into 215xxx.zip and upload the 215xxx.zip on moodle.