**Tree**

Colab Link: [Copy of Lab08-Tree.ipynb](https://colab.research.google.com/drive/1upPMwVQ8mgdpf_fbWU9lQiCThy6PbXie?usp=sharing)

Instructions for students:

● Complete the following methods on Tree.

● You may use any language to complete the tasks.

● All your methods must be written in one single .java or .py or .pynb file. DO NOT CREATE separate files for each task.

● If you are using JAVA, you must include the main method as well which should test your other methods and print the outputs according to the tasks.

● If you are using PYTHON, then follow the coding templates shared in this folder.

NOTE:

**● YOU CANNOT USE ANY BUILT-IN FUNCTION EXCEPT** len **IN PYTHON. [negative indexing, append is prohibited]**

**● YOU HAVE TO MENTION SIZE OF ARRAY WHILE INITIALIZATION**

* **DO NOT USE LIST**

## Mirror Mirror:

Given a binary tree, convert it into its mirror.

Sample Input:

10

/ \

20 30

/ \

40 60

Sample Output:

10 10

/ \ Mirror / \

20 30 —> 30 20

/ \ / \

40 60 60 40

Inorder Traversal of mirror: 30 10 60 20 40

## Level Max:

Given a binary tree, find the largest value in each level.

Sample Input:

4

/ \

2 9

/ / \

7 3 5

Sample Output: 4 9 7

Explanation:

There are 3 levels in the tree

Level 0: {4}, max = 4

Level 1: {2, 9}, max = 9

Level 2: {3, 5, 7}, max = 7

## Inorder Successor:

Given a BST, and a reference to a Node x in the BST, find the Inorder Successor of the given node in the BST.

**DO NOT USE LIST**

Sample Input:

20

/ \

8 22

/ \

4 12

/ \

10 14

x = reference of the node containing 8

Sample Output: reference of the node 10

Explanation:

The inorder successor of a parent node is the smallest (leftmost) node in the right subtree. The leftmost node in the right subtree of parent node 8 is 10.

Another explanation is that, the inorder traversal of the given tree:

4 8 10 12 14 20 22

Hence, the inorder successor of 8 is 10.

## Kth Smallest:

Given a Binary search tree, your task is to complete the function which will return the Kth smallest element without doing any modification in the Binary Search Tree.

**DO NOT USE LIST**

Sample Input:

20

/ \

8 22

/ \

4 12

/ \

10 14

k = 2

Sample Output: 8

## Ungraded Bonus Task:

Given a Binary Tree, Write a function that finds the difference between sum of all nodes present at odd and even levels in a binary tree, i.e. sum of all even level nodes - sum of all odd level nodes.

| Sample Input: | Sample Output | Explanation |
| --- | --- | --- |
|  | -4 | 1-2-3+4+5+6-7-8 = -4 |