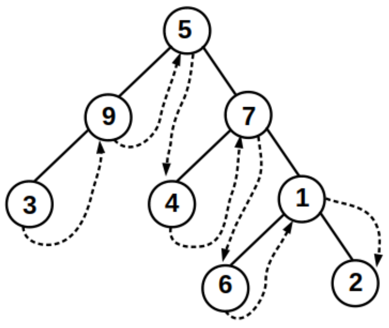
Algorithms Laboratory (CS29203)
Assignment 5: Tree data structure
Department of CSE, IIT Kharagpur

13th October 2022

Question-1 (50 points)

Consider a binary tree where each node contains an integer type data and two pointers *left and *right that points to the left and right child, respectively. Now imagine that each node contains an extra pointer *inOrderSuccessor that points to the inorder successor of the node. For example, consider the following tree where the dotted line represents the extra pointer to the inorder successor of each node.



Your task is to first construct a tree from a given set of data (create the nodes by dynamic memory allocation). For example, the above tree can be constructed from the following sequence of pseudocode:

```
Node* root = createNode(5);
root->left = createNode(9);
root->right = createNode(7);
root->left->left = createNode(3);
root->right->left = createNode(4);
root->right->right = createNode(1);
root->right->right->left = createNode(6);
root->right->right->right = createNode(2);
```

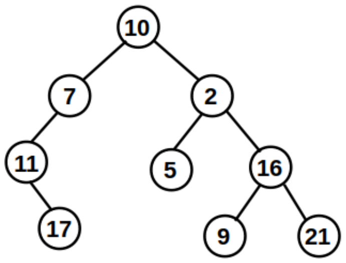
Finally print the inorder successor of each node in the tree using the extra pointer *inOrderSuccessor. That is, you will start from the leftmost node and traverse to the consecutive inorder successor using the extra pointer. Note that you will not get any credit for simply printing the inorder traversal (and not using the idea of the extra pointer).

Example: (considering the same tree as in the figure)
Output:

```
The inorder successor of 3 is 9
The inorder successor of 9 is 5
The inorder successor of 5 is 4
The inorder successor of 4 is 7
The inorder successor of 7 is 6
The inorder successor of 6 is 1
The inorder successor of 1 is 2
```

Question-2 (50 points)

Consider a binary tree that stores integer data values. We wish to compute the reachability of different nodes in the tree from the leaf nodes. In order to do this, we will list the nodes which are directly reachable from the leaf nodes, then list the nodes which are 1-hop distance away from the leaf nodes (i.e. there will be an intermediate node along the path), then list the nodes which are 2-hop distance away from the leaf nodes (i.e. there will be two intermediate nodes), and so on. For example, consider the following tree:



In this tree, the nodes which are directly reachable (0-hop away) from the leaf nodes are 11, 2, 16. The nodes which are 1-hop away from the leaf nodes are 7, 10, 2. Similarly the node which is 2-hop away from the leaf nodes is 10. Your task is to implement this and print the nodes which are at different hop away from the leaf nodes. The program should consider all the hops possible upto the root node.

Example: (considering the same tree as in the above figure)
Output:

```
0-hop away nodes from the leaf nodes: 11, 2, 16
1-hop away nodes from the leaf nodes: 7, 10, 2
2-hop away nodes from the leaf nodes: 10
```