Data Structures Lab Fall 2023
Lab 10
Binary Trees Implementation

Task One:

Implement a binary tree through array-based implementation. You must make the following classes and functions:

1. You will make a Binary tree class with the following attributes.

```
Class BinaryTree {
  Int * treeArray
  Int capacity
  Int root
  Int current_index
}
```

2. ***BinaryTree():*** You need to make a constructor that allocates memory to the tree array. Then you should initialize the array with -1, denoting that all elements are currently empty. Set your root to 0 and start your current index from 0.

3. ***Void insert():*** Insert the new value into the array at the current index and then increment the current index.

4. ***Void inorderTraversal(int index***) : Function to perform inorder traversal. Note that this function will be called recursively.

   Base Case: if treearray[index] == -1
   Recursive Call left (2*index+1)
   Print
   Recursive Call right (2*index+2)

5. ***Void displayInorder():*** Function that will display the inorder traversal. Just call the recursive function with root.

You can use Queue standard library

Task Two:

Write a C++ code to implement the node-based implementation of a binary tree. You must implement the following functions.

1. You will need to make a node class that has the following parameters.

   Class Node {
   Int data;
   Node * left;
   Node * right;
   }

2. Now you need to make a Binary tree class that will contain the root of the tree.

   Class BinaryTree{
   Node * Root;
   }

3. ***Void insert(int data):***

   You can insert the node in the following manner if you want to insert the node of the left child.
   Following is an example of how you will add a node to left of root node

   //Step 1: Make a new tree node;
   Node * newNode = new Node(data);
   //Step 2: Make current node that points to root
   Node * currentnode = new Node(data);
   //Step 3: Adjust the new node to the left of the current node that is pointing to root
   current->left = newNode

   The tricky part in this question is that you have to add nodes in such a way that a complete binary tree is formed.

4. ***Void InoderTraversal(Node * treeNode):***

   This is a recursive function that will be called to traverse the tree in order.

5. ***Void Display()***

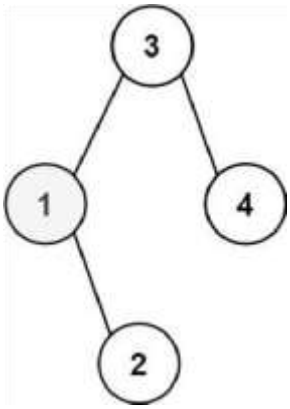
   This function will just call inoderTraversal function with root of tree

You can use Stack standard library

Task Three:

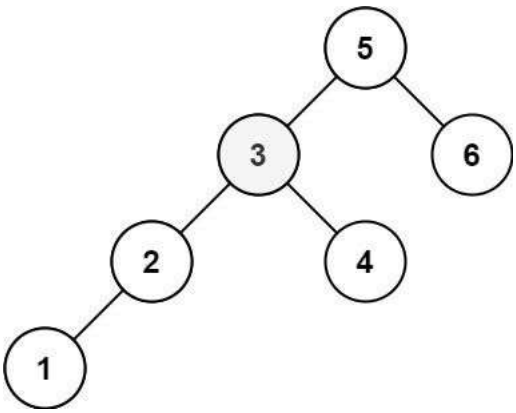
Given the root of a binary tree and an integer k, return *the* k^{th} *smallest value (**1-indexed**) of all the values of the nodes in the tree.* You have to use DFS to find the k^{th} smallest element.

Example 1:



Input: root = [3,1,4,null,2], k = 1
Output: 1

Example 2:



Input: root = [5,3,6,2,4,null,null,1], k = 3
Output: 3

HOME TASKS (Non Graded)

- Write a C++ code to make array based tree and write a function that checks if it is complete binary tree
- Write a C++ code to make array based tree and write a function that checks if it is a balanced tree
- Write a C++ code to make array based tree and write a function that checks if it is full binary tree
- Write a C++ code to make array based tree and write a function that checks if it is complete binary tree