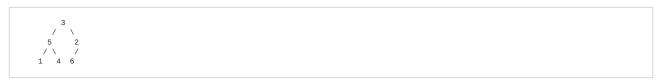


Day 23: Review+Binary Trees!



Welcome to Day 23! Review everything we've learned so far by making Tic Tac Toe from scratch with this video, or just jump right into the problem.

You are given a pointer root pointing to the root of a binary tree. You need to print the level order traversal of this tree. In level order traversal, we visit the nodes level by level from left to right. For example:



For the above tree, the level order traversal is $3 \rightarrow 5 \rightarrow 2 \rightarrow 1 \rightarrow 4 \rightarrow 6$.

HINT: A queue could be helpful.

The code for input/output is already handled in the editor. You only have to complete the function levelOrder given in the editor.

Good luck!

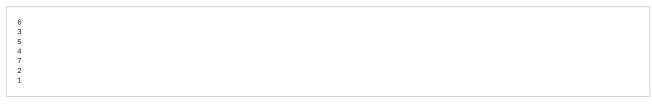
Input Format

First line contains T, the number of test cases. Next T lines contain an integer data to be added to the binary search tree.

Output Format

Print the values of the level order traversal separated by spaces.

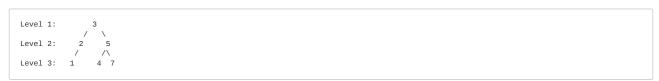
Sample Input



Sample Output

```
3 2 5 1 4 7
```

Explanation



We need to print the nodes level by level. We process each level from left to right.

Level Order Traversal: 3 -> 2 -> 5 -> 1 -> 4 -> 7

Submissions: 3106 Max Score: 100 Difficulty: Easy More C++ Current Buffer (saved locally, editable) & 49 Ö 1 v #include <iostream> #include <cstddef> 3 #include <queue> #include <string> #include <cstdlib> using namespace std; 8 class Node{ 9 public: int data;
Node *left,*right; 10 11 Node(int d){ 12 ▼ 13 data=d; 14 left=right=NULL; 15 16 17 class Solution{ public: 18 Node* insert(Node* root, int data){ 19 ▼ 20 ▼ if(root==NULL){ 21 return new Node(data); 22 23 ▼ else{ Node* cur; 24 25 ▼ if(data<=root->data){ cur=insert(root->left,data); 26 27 root->left=cur; 28 29 י else{ 30 31 cur=insert(root->right,data); root->right=cur; 32 33 return root; 34 35 void levelOrder(Node * root){ 36 ▼ 37 //Write your code here 38 39 41 };//End of Solution 42 **v** int main(){ Solution myTree; Node* root=NULL; int T,data; 44 45 46 cin>>T; while(T-->0){ 47 ▼ 48 cin>>data; 49 root= myTree.insert(root,data); 50 51 myTree.levelOrder(root); 52 53 } return 0; Line: 7 Col: 1 Test against custom input **1** Upload Code as File Run Code Submit Code

Join us on IRC at #hackerrank on freenode for hugs or bugs.

Contest Calendar | Blog | Scoring | Environment | FAQ | About Us | Support | Careers | Privacy Policy | Request a Feature