Left View of Binary Tree

Easy Accuracy: 37.86% Submissions: 100k+ Points: 2

Given a Binary Tree, print Left view of it. Left view of a Binary Tree is set of nodes visible when tree is visited from Left side. The task is to complete the function **leftView()**, which accepts root of the tree as argument.

Left view of following tree is 1 2 4 8.

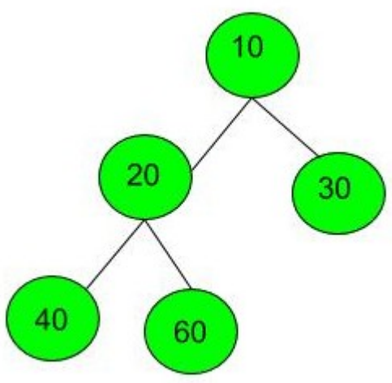
```
    1
  /   \
 2     3
/  \  /  \
4  5 6   7
 \
  8
```

Example 1:

```
Input:
   1
 /  \
3    2
Output: 1 3
```

Example 2:

```
Input:
```




```
Output: 10 20 40
```

Your Task:
You just have to **complete** the function **leftView()** that prints the left view. The newline is automatically appended by the driver code.
Expected Time Complexity: O(N).
Expected Auxiliary Space: O(Height of the Tree).

Constraints:
0 <= Number of nodes <= 100
1 <= Data of a node <= 1000

View Bookmarked Problems (https://practice.geeksforgeeks.org/explore/?problemType=bookmark)

Company Tags

Topic Tags

Related Courses

Related Interview Experiences

We are replacing the old Disqus forum with the new Discussions section given below. Click here (https://practice.geeksforgeeks.org/comments/left-view-of-binary-tree/1/?rel=https://practice.geeksforgeeks.org/problems/left-view-of-binary-tree/1) to view old Disqus comments.

Output Window

You get marks only for the first correct submission if you solve the problem without viewing the full solution.

Problem Solved Successfully

Test Cases Passed: **603**/ 603

Total Time Taken: **0.2**/1.42

Correct Submission Count: **4**

Attempts No.: **6**

```
132  void solve(Node* root, vector<int>& ans, int level){
133      if(root == NULL)
134          return;
135      if(level >= ans.size())
136          ans.push_back(root->data);
137
138      solve(root->left, ans, level + 1);
139      solve(root->right, ans, level + 1);
140      return;
141  }
142  vector<int> leftView(Node *root)
143  {
144      // Your code here
145      vector<int> ans;
146      if(root == NULL)
147          return ans;
148      solve(root, ans, 0);
149      return ans;
150  }
151
```

Discussions

bipulharsh123 4 days ago

+1

```
void getLeftView(Node *ptr, vector<int>&result, i
  if(!ptr)
    return;
  if(level > result.scd ize())
    result.push_back(ptr->data);
  getLeftView(ptr->left, result, level+1);
  getLeftView(ptr->right, result, level+1);
}
//Function to return a list containing elements of l
```

Average Time: 20m
Your Time: 16m

Compile & Run Submit