[LeetCode] Binary Tree Level Order Traversal 二叉树层次遍历(DFS | BFS)

原创 Eastmount 2015-09-11 02:20:37 5263 收藏 版权

分类专栏： LeetCode 文章标签： leetcode 二叉树 遍历 dfs bfs

目录：

题目概述：

Given a binary tree, return the level order traversal of its nodes' values. (ie, from left to right, level by level).

For example:

Given binary tree `{3,9,20,#,#,15,7}`,

```
    3
   / \
  9  20
    /  \
   15   7
```

return its level order traversal as:

```
[
  [3],
  [9,20],
  [15,7]
]
```

Here's an example:

```
  1
 / \
2   3
   /
  4
   \
    5
```

The above binary tree is serialized as "{1,2,3,#,#,4,#,#,5}".

题目分析：

本题考查的就是二叉树的层次遍历，需要注意的是二叉树用数组的表示方法，二叉树的每层是从左到右存入数组的。方法包括：

1.层次遍历。二维数组存储数字和深度，输出二维数组即可，过于复杂。

2.通过队列BFS广度优先搜索。

3.通过DFS深度优先搜索实现。

我的代码：

```
/**
 * Definition for a binary tree node.
 * struct TreeNode {
 *     int val;
 *     TreeNode *left;
 *     TreeNode *right;
 *     TreeNode(int x) : val(x), left(NULL), right(NULL) {}
 * };
 */
class Solution {
public:
    //二叉树层次遍历 通过队列BFS广度优先搜索
    vector<vector<int>> levelOrder(TreeNode* root) {
        vector<vector<int>> result;
        queue<TreeNode*>q;
        vector<int> level;           //每层结果
        int size,i;
        TreeNode* p;

        if(root==NULL) return result;
        q.push(root);                //入队
        while(!q.empty()) {
            //队列中有几个元素就依次遍历每个元素的左右结点
            level.clear();
            size=q.size();
```