

2641.二叉树的堂兄弟节点 II

(这题完全是困难的难度吧，怎么标的是中等啊？)

题目大意

二叉树的每一个节点存了一个值，现在把每一个节点的值同时替换为“**同一层节点的值之和减去它和它兄弟节点值的和**”。

题目分析

树上节点个数为 $n \leq 1e5$ ，不能使用朴素的 $id, 2 * id, 2 * id + 1$ 的方式来对节点进行编号，因此我们考虑进行动态编号。题目给的接口极其阴间，因此我们在编号和节点之间建立一个映射：

```
unordered_map<int, TreeNode*> access;
```

我们一共需要如下信息：

1. 每一层的节点值之和（处理出每一层具体有哪些节点）
2. 每个节点兄弟结点的编号

同时，拷贝每个节点改变前的值，用容器记录下以上信息：

```
unordered_map<int, int> value; //original value
unordered_map<int, int> bro;    //id of brother
vector<int> exist[100005];      //node id from every layer
```

通过一次 BFS 处理出这些信息：

```
int max_dep = -1, max_node, tot = 1;
struct INFO{
    TreeNode* now;
    int dep;
    int bro_id;
    int dir;
};
void bfs(TreeNode* root){
    queue<INFO> search;
    search.push({root, 0, 0, 1});
    while(!search.empty()){
```

```

INFO front = search.front();
search.pop();
TreeNode* now = front.now;
int dep = front.dep, bro_id = front.bro_id;
if(!now) return;
int dir = front.dir;
exist[dep].push_back(dir);
max_dep = max(max_dep, dep);
access[dir] = nbow, value[dir] = now->val, bro[dir] = bro_id;
if(now->left){
    if(now->right) search.push({now->left, dep + 1, tot + 2, tot + 1});
    else search.push({now->left, dep + 1, 0, tot + 1});
}
if(now->right){
    if(now->left) search.push({now->right, dep + 1, tot + 1, tot + 2});
    else search.push({now->right, dep + 1, 0, tot + 1});
}
if(now->left) tot++;
if(now->right) tot++;
}
}

```

考虑到 BFS 搜索顺序的特性，如果左儿子节点 id 的右兄弟节点存在，则其编号为 $id + 1$ ，使用结构体 *INFO* 维护 BFS 信息。

经过预处理后，我们已经获取了每一层都有哪些节点，只需要对每一层进行循环，求出这一层节点值之和，随后将每一个节点修改为“**同一层节点的值之和减去它和它兄弟结点值的和**”即可，时间复杂度为 $O(n)$ 。代码片段如下：

```

TreeNode* replaceValueInTree(TreeNode* root) {
    bfs(root);
    int level_sum, subtract;
    for(int i=0; i<=max_dep; i++){
        level_sum = 0;
        for(auto ele : exist[i])
            level_sum += access[ele]->val;
        for(auto ele : exist[i]){
            subtract = value[ele];
            if(bro[ele])subtract += value[bro[ele]];
            access[ele]->val = level_sum - subtract;
        }
    }
    return root;
}

```

写在最后

实际上，可以把 `vector<int> exist` 替换为 `vector<TreeNode*> exist`，这样可以避免麻烦的动态开点，代码可读性也会变高，在 BFS 时只需要 push 节点就可以了。一开始写的时候脑子晕了（