

N 叉树的前序遍历

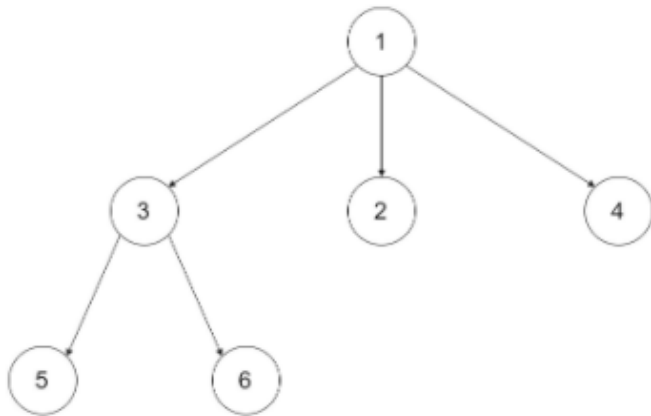
给定一个 N 叉树，返回其节点值的 **前序遍历**。

N 叉树 在输入中按层序遍历进行序列化表示，每组子节点由空值 null 分隔（请参见示例）。

进阶：

递归法很简单，你可以使用迭代法完成此题吗？

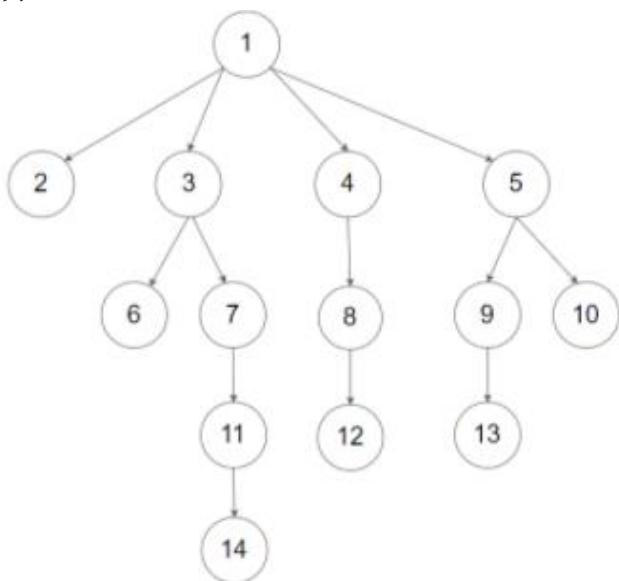
示例 1：



输入：root = [1,null,3,2,4,null,5,6]

输出：[1,3,5,6,2,4]

示例 2：



输入: root =

[1,null,2,3,4,5,null,null,6,7,null,8,null,9,10,null,null,11,null,12,null,13,null,null,14]

输出: [1,2,3,6,7,11,14,4,8,12,5,9,13,10]

递归:

```
/*
// Definition for a Node.
class Node {
public:
    int val;
    vector<Node*> children;

    Node() {}

    Node(int _val) {
        val = _val;
    }

    Node(int _val, vector<Node*> _children) {
        val = _val;
        children = _children;
    }
};
*/

class Solution {
public:
    void travel(Node* node, vector<int>& res)
    {
        if(node==nullptr) return;
        res.push_back(node->val);
        for(int i=0;i<node->children.size();i++)
        {
            travel(node->children[i],res);
        }
    }
    vector<int> preorder(Node* root) {
        vector<int> res;
        if(root==nullptr)
        {
            return res;
        }
    }
};
```

```
        travel(root,res);
        return res;
    }
};
```

执行结果: **通过** [显示详情 >](#)

执行用时: **24 ms** , 在所有 C++ 提交中击败了 **49.23%** 的用户

内存消耗: **11 MB** , 在所有 C++ 提交中击败了 **94.81%** 的用户

炫耀一下:



[写题解, 分享我的解题思路](#)

迭代:

首先把根节点入栈,因为根节点是前序遍历中的第一个节点。随后每次我们从栈顶取出一个节点 **u**,它是我们当前遍历到的节点,并把 **u** 的所有子节点逆序推入栈中。例如 **u** 的子节点从左到右为 **v1,v2,v3**,那么推入栈的顺序应当为 **v3,v2,v1**,这样就保证了下一个遍历到的节点出现在栈顶的位置。

```
class Solution {
public:
    vector<int> preorder(Node* root) {
        vector<int> res;
        if(root==nullptr)
        {
            return res;
        }
        stack<Node*> st;
        st.push(root);
        while(!st.empty())
        {
            Node* node=st.top();
            st.pop();
            if(node)
            {
```

```
        res.push_back(node->val);
    }
    else
    {
        continue;
    }
    if(!node->children.empty())
    {
        for(int i=node->children.size()-1;i>=0;i--)//逆序压栈
        {
            st.push(node->children[i]);
        }
    }
}
return res;
}
};
```


执行结果: **通过** [显示详情 >](#)

执行用时: **20 ms** , 在所有 C++ 提交中击败了 **76.51%** 的用户

内存消耗: **11.3 MB** , 在所有 C++ 提交中击败了 **45.93%** 的用户

炫耀一下:



 写题解, 分享我的解题思路