

```
/*
给定一个 N 叉树，返回其节点值的后序遍历。
```

例如，给定一个 3叉树：

```
      1
     / \ \
    3  2  4
   / \
  5  6
返回其后序遍历：[5,6,3,2,4,1]。
```

说明：递归法很简单，你可以使用迭代法完成此题吗？

来源：力扣 (LeetCode)

链接：<https://leetcode-cn.com/problems/n-ary-tree-postorder-traversal>

著作权归领扣网络所有。商业转载请联系官方授权，非商业转载请注明出处。

```
*/
```

分析：

- N叉树的后序遍历(左右根)-递归法
- N叉树的后序遍历(左右根)-迭代法:(根右左+翻转)

方法一:递归法

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

    Node() {}

    Node(int _val, vector<Node*> _children) {
        val = _val;
        children = _children;
    }
};
*/
class Solution
{
private:
    vector<int> ret_val;

    void __PostOrder(Node* node)
```

```

    {
        int size = 0;
        int i = 0;
        if(node==NULL)
        {
            return;
        }
        else
        {
            size = node->children.size();
            for(i=0;i<size;i++)
            {
                __PostOrder(node->children[i]);
            }
            ret_val.push_back(node->val);
        }
    }

public:
    vector<int> postorder(Node* root)
    {
        ret_val.clear();
        __PostOrder(root);
        return ret_val;
    }
};

```

/*
 执行结果:
 通过
 显示详情
 执行用时 :216 ms, 在所有 C++ 提交中击败了84.34% 的用户
 内存消耗 :32.7 MB, 在所有 C++ 提交中击败了78.75%的用户
 */

方法二:C++ 迭代法

```

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

    Node() {}

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

```

```

*/
class Solution
{
public:
    vector<int> postorder(Node* root)
    {
        vector<int>    ret_val          ;
        stack<Node*>    sn                ;
        Node*          temp    =    NULL    ;
        int             i          =    0          ;
        int             size       =    0          ;

        if(root!=NULL)
        {
            sn.push(root);
            while(!sn.empty())
            {
                temp = sn.top();
                sn.pop();
                ret_val.push_back(temp->val);
                size = temp->children.size();
                for(i=0;i<size;i++)
                {
                    sn.push(temp->children[i]);
                }
            }

            reverse(ret_val.begin(), ret_val.end());
            return ret_val;
        }
    };
};

```

/*

执行结果:

通过

[显示详情](#)

执行用时 :208 ms, 在所有 C++ 提交中击败了94.31% 的用户

内存消耗 :32.9 MB, 在所有 C++ 提交中击败了66.88%的用户

*/