

```
/*
给定一个 N 叉树，返回其节点值的前序遍历。
```

例如，给定一个 3叉树：

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

返回其前序遍历：[1,3,5,6,2,4]。

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-preorder-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叉树的前序遍历(根左右)-递归法
- N叉树的前序遍历(根左右)-迭代法,需要辅助栈空间保存当前根节点

方法一:C++,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 __PreOrder(Node* node)
    {
        int size = 0;
```

```

        int i = 0;
        if(node==NULL)
        {
            return;
        }
        else
        {
            ret_val.push_back(node->val);
            size = node->children.size();
            for(i=0;i<size;i++)
            {
                __PreOrder(node->children[i]);
            }
        }
    }
public:
    vector<int> preorder(Node* root)
    {
        ret_val.clear();
        __PreOrder(root);
        return ret_val;
    }
};
/*
执行结果:
通过
显示详情
执行用时 :368 ms, 在所有 C++ 提交中击败了18.87% 的用户
内存消耗 :32.7 MB, 在所有 C++ 提交中击败了84.08%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
{
private:
    vector<int> ret_val;

```

```

public:
    vector<int> preorder(Node* root)
    {
        vector<int>    ret_val    ;
        stack<Node*>    sn        ;
        Node*          temp      ;
        int             size = 0   ;
        int             i       = 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=size-1;i>=0;i--)
                {
                    sn.push(temp->children[i]);
                }
            }
        }
        return ret_val;
    }
};

```

/*

执行结果:

通过

[显示详情](#)

执行用时 :372 ms, 在所有 C++ 提交中击败了17.57% 的用户

内存消耗 :32.9 MB, 在所有 C++ 提交中击败了71.23%的用户

*/