# 题目

输入一个链表的头节点，从尾到头反过来返回每个节点的值（用数组返回）。

**示例 1：**

输入：head = [1,3,2]

输出：[2,3,1]

**限制：**

0 <= 链表长度 <= 10000

# 分析

## 方法一：递归法

**思路：**

递归输出后面一个节点，再输出当前节点

**代码：**

/\*\*

\* Definition for singly-linked list.

\* struct ListNode {

\* int val;

\* ListNode \*next;

\* ListNode(int x) : val(x), next(NULL) {}

\* };

\*/

class Solution {

public:

//方法1 递归

vector<int> reversePrint(ListNode\* head) {

vector<int>res;

helper(head,res);

return res;

}

void helper(ListNode\* head,vector<int>& res){

if(head){

if(head->next){

helper(head->next,res);

}

res.push\_back(head->val);

}

return ;

}

};

## 方法二：栈

**思路：**

栈依次存入节点值，再依次取出实现逆序

**代码：**

/\*\*

\* Definition for singly-linked list.

\* struct ListNode {

\* int val;

\* ListNode \*next;

\* ListNode(int x) : val(x), next(NULL) {}

\* };

\*/

class Solution {

public:

vector<int> reversePrint(ListNode\* head) {

vector<int> ret;

stack<int> s;

while(head)

{

s.push(head->val);

head = head->next;

}

while(!s.empty())

//不能使用for(int i=0;i<stk.size();i++)

{

ret.push\_back(s.top());

s.pop();

}

return ret;

}

};