

# 206-反转链表

## 题述

206. 反转链表

难度 简单

👍 2166

☆

📄

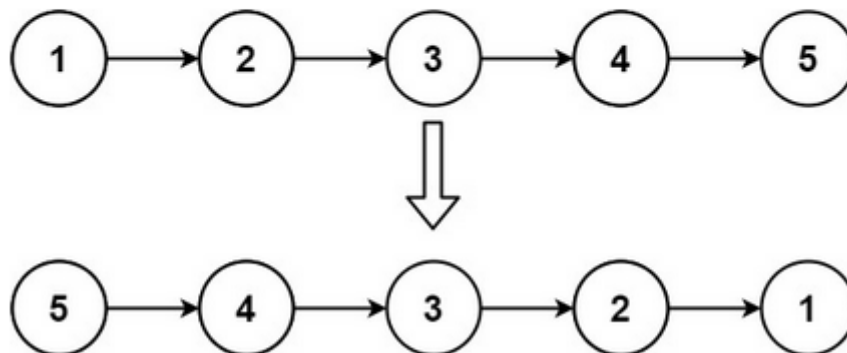
🔍

🔔

💬

给你单链表的头节点 `head`，请你反转链表，并返回反转后的链表。

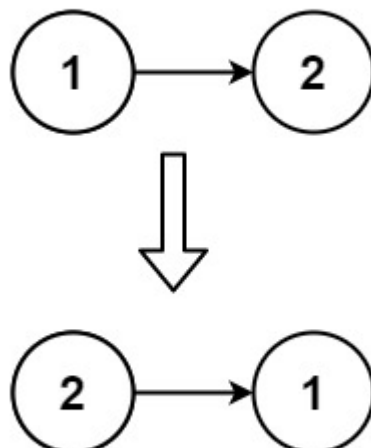
示例 1:



输入: `head = [1,2,3,4,5]`

输出: `[5,4,3,2,1]`

示例 2:



输入: `head = [1,2]`

输出: `[2,1]`

## 思考

暴力解法就是建立一个新链表 然后依次将元素从尾到头给存放进去

但是这样做造成了对内存空间的浪费

其实不妨换位思考一下 反转链表的本质其实可以理解为将每一个结点之间的指针域next的指向反转即可

那么我们就可以使用双指针法来进行解决这道题

## 题解

### 双指针法

```
/**
 * Definition for singly-linked list.
 * struct ListNode {
 *     int val;
 *     ListNode *next;
 *     ListNode() : val(0), next(nullptr) {}
 *     ListNode(int x) : val(x), next(nullptr) {}
 *     ListNode(int x, ListNode *next) : val(x), next(next) {}
 * };
 */
class Solution {
public:
    ListNode* reverseList(ListNode* head)
    {
        //反转链表 改变next指针指向即可
        //双指针法
        ListNode* temp; //保存当前结点的下一结点
        ListNode* cur=head;
        ListNode* pre=nullptr;
        while(cur!=nullptr)
        {
            temp=cur->next;    //暂存p的下一结点因为接下来要改变cur->next
            cur->next=pre;    //实现翻转操作

            //更新pre和cur等工作指针
            pre=cur;
            cur=temp;
        }
        return pre;
    }
};
```

### 递归法

```
/**
 * Definition for singly-linked list.
 * struct ListNode {
 *     int val;
 *     ListNode *next;
 *     ListNode() : val(0), next(nullptr) {}
 *     ListNode(int x) : val(x), next(nullptr) {}
 *     ListNode(int x, ListNode *next) : val(x), next(next) {}
 * };
 */
class Solution {
public:
    ListNode* reverse(ListNode* pre,ListNode* cur)
    {
        //递归算法的轮子
        if(cur==nullptr)
```

```

{
    //递归终止条件
    return pre;
}
ListNode* temp=cur->next;
cur->next=pre;

//递归算法的思路和双指针法相同
//其实就是递归代替了 pre=cur; cur=temp; 两步
return reverse(cur,temp);
}

ListNode* reverseList(ListNode* head)
{
    //和双指针法初始化是一样的逻辑
    return reverse(nullptr,head);
}
};

```

执行结果: **通过** [显示详情](#)

[添加备注](#)

执行用时: **4 ms** , 在所有 C++ 提交中击败了 **90.45%** 的用户

内存消耗: **8 MB** , 在所有 C++ 提交中击败了 **86.86%** 的用户

通过测试用例: **28 / 28**

炫耀一下:



[写题解，分享我的解题思路](#)

提交结果	执行用时	内存消耗	语言	提交时间	备注
<b>通过</b>	4 ms	8 MB	C++	2021/12/27 14:08	<a href="#">添加备注</a>
<b>通过</b>	0 ms	8.2 MB	C++	2021/09/03 15:10	<a href="#">添加备注</a>

## 总结

拿到题目先思考

别直接拿到题目就像着使用暴力解法

不妨思考一下双指针啊 递归啊之类的方法可不可行

