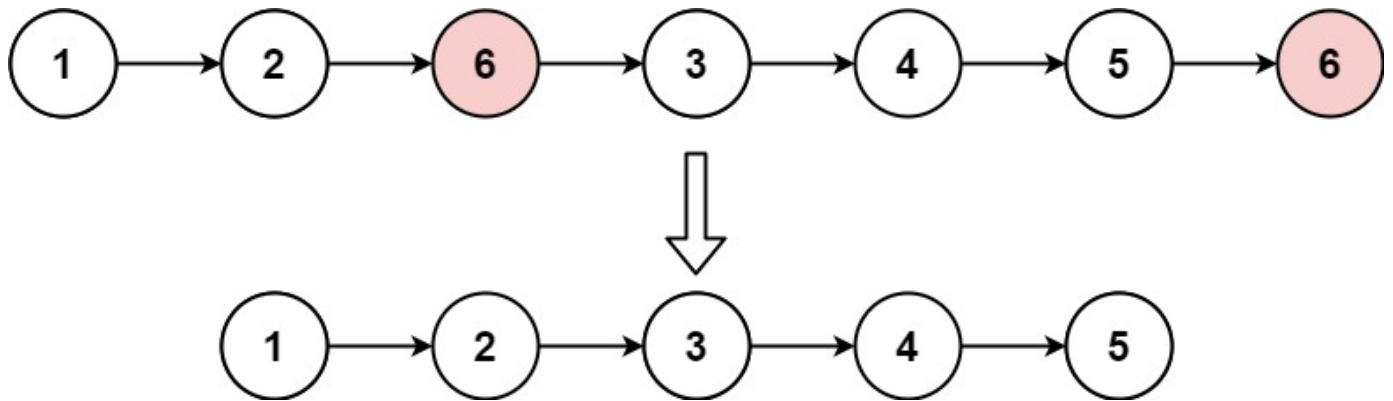


## 第34题——移除链表元素

给你一个链表的头节点 `head` 和一个整数 `val`，请你删除链表中所有满足 `Node.val == val` 的节点，并返回 新的头节点。

示例1:



输入: `head = [1,2,6,3,4,5,6]`, `val = 6`

输出: `[1,2,3,4,5]`

示例 2:

输入: `head = []`, `val = 1`

输出: `[]`

示例 3:

输入: `head = [7,7,7,7]`, `val = 7`

解题思我们使用迭代法,遍历整个链表,需要用一个结点来保存head,每当发现当前这个结点的值=`val`时,那么我们让这个结点的上一个结点的指针指向这个结点所指向的内容,如此一来,我们就把该结点从这个链表中剔除了;

当然,因为我们使用的时C/C++语言来书写,怎么能忘记内存释放呢,我们在剔除结点的同时应该把它的内存空间释放掉.

## 双指针(1)

代码如下:

```
/**
 * 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* removeElements(ListNode* head, int val) {
        ListNode *new_head=new ListNode();
        new_head->next=head;
        ListNode *he=new_head;
        while(he->next!=NULL)
        {
            if(he->next->val==val)
            {
                ListNode *temp=he->next;//创建临时指针指向该结点
                he->next=he->next->next;
                delete temp;//删除该结点
            }
            else
            {
                he=he->next;
            }
        }
        return new_head->next;
    }
};
```

作者: sakura7301

链接: <https://leetcode-cn.com/problems/remove-linked-list-elements/solution/shan-chu-jie-dian-by-sakura7301-tnx/>

来源: 力扣 (LeetCode)

著作权归作者所有。商业转载请联系作者获得授权, 非商业转载请注明出处。