# 题目

# 分析

## 方法一：递归法

class Solution {

public:

ListNode\* removeElements(ListNode\* head, int val) {

if (!head)

return head;

head->next = removeElements(head->next, val);

return head->val == val ? head->next : head;

}

};

## 方法二：原链表

class Solution {

public:

ListNode\* removeElements(ListNode\* head, int val) {

ListNode \*p=head;//当前判断元素

ListNode \*k=p;//指向前一个元素

while(p)

{

if(p->val==val)

{

if(p==head)//头元素要删的情况

{

head=p->next;

p=head;

if(head==NULL)

return head;

else continue;

}

else if(p->next==NULL)//末尾元素要删的情况

{

k->next=NULL;

return head;

}

else//中间元素

{

k->next=p->next;

p=k->next;

continue;

}

}

k=p;

p=p->next;

}

return head;

}

};

## 方法三：新链表

class Solution {

public:

ListNode\* removeElements(ListNode\* head, int val) {

ListNode \*sentinel=new ListNode(INT\_MAX);

sentinel->next=head; //构造头结点

ListNode \*p=sentinel,\*temp;

while(p->next){

if(p->next->val==val){

temp=p->next;

p->next=p->next->next;

delete temp;

}

else

p=p->next;

}

return sentinel->next;

}

};

复杂度分析：

时间复杂度：O(N)，只遍历了一次。

空间复杂度：O(1)。