203 Remove Linked List Elements

2018年4月4日 16:58

Question:

Remove all elements from a linked list of integers that have value ***val***.

Example

Given: 1 --> 2 --> 6 --> 3 --> 4 --> 5 --> 6, ***val*** = 6

Return: 1 --> 2 --> 3 --> 4 --> 5

来自 <https://leetcode.com/problems/remove-linked-list-elements/description/>

删除链表中等于给定值 *val* 的所有元素。

示例

给定: 1 --> 2 --> 6 --> 3 --> 4 --> 5 --> 6, *val* = 6

返回: 1 --> 2 --> 3 --> 4 --> 5

Solution for Python3:

```
1  # Definition for singly-linked list.
2  # class ListNode:
3  #     def __init__(self, x):
4  #         self.val = x
5  #         self.next = None
6  # 栈太深，溢出
7  class Solution:
8      def removeElements(self, head, val):
9          """
10         :type head: ListNode
11         :type val: int
12         :rtype: ListNode
13         """
14        if not head:
15            return head
16        head.next = self.removeElements(head.next, val);
17        return head.next if head.val == val else head
18
19 class Solution:
20     def removeElements(self, head, val):
21         """
22         :type head: ListNode
23         :type val: int
```

```
        :rtype: ListNode
        """
        dummy = ListNode(-1)
        dummy.next = head
        P = dummy
        while P.next:
            if P.next.val == val:
                P.next = P.next.next
            else:
                P = P.next
        return dummy.next
```

Solution for C++:

```
/**
 * Definition for singly-linked list.
 * struct ListNode {
 *     int val;
 *     ListNode *next;
 *     ListNode(int x) : val(x), next(NULL) {}
 * };
 */
class Solution {
public:
    ListNode* removeElements(ListNode* head, int val) {
        if (head == NULL) {
            return head;
        }
        head->next = removeElements(head->next, val);
        return head->val == val ? head->next : head;
    }
};

class Solution2 {
public:
    ListNode* removeElements(ListNode* head, int val) {
        ListNode* dummy = new ListNode(-1);
        dummy->next = head;
        ListNode* p = dummy;
        while (p->next) {
            if (p->next->val == val) {
```

```
28                  p->next = p->next->next;
29              } else {
30                  p = p->next;
31              }
32          }
33          return dummy->next;
34      }
35  };
```