

## 86. Partition List

题目描述: <https://leetcode.com/problems/partition-list/>

给定一个链表，和一个整数target，要求按照把target小的放在前半部分，target大或等于的放在后半部分，要求每部分内的相对顺序不变，例如：

```
Given 1->4->3->2->5->2  x = 3,
return 1->2->2->4->3->5
```

解题思路：

找到第一个比target大或等于的元素n，从这个地方的前一个pre分界，先遍历前面找到属于后面的节点放在pre后面，然后再遍历后半部分，找到比target小的，放到前面。

代码：

```
/**
 * Definition for singly-linked list.
 * struct ListNode {
 *     int val;
 *     ListNode *next;
 *     ListNode(int x) : val(x), next(NULL) {}
 * };
 */
class Solution {
public:
    ListNode* partition(ListNode* head, int x) {
        if(head == NULL || head->next == NULL){
            return head;
        }
        ListNode *p, *n, *l;
        ListNode* fakeHead = new ListNode(0);
        fakeHead->next = head;
        ListNode *pre = fakeHead;
        pre = fakeHead;
        n = head;
        while(n != NULL && n->val < x){
            pre = n;
            n = n->next;
        }
        if(n == NULL)
            return head;
    }
};
```

```

cout << "pre->val:" << pre->val << endl;
pre = fakeHead;
l = n;
p = head;
while(p != n){
    if(p->val >= x){
        ListNode * t1 = l->next;
        ListNode * t2 = p->next;
        l->next = p;
        p->next = t1;
        pre->next = t2;
        l = p;
        p = t2;
    }
    else{
        pre = p;
        p = p->next;
    }
}
p = pre;
l = n->next;
pre = n;
while(l){
    if(l->val < x){
        ListNode *t1 = p->next;
        ListNode *t2 = l->next;
        p->next = l;
        l->next = t1;
        pre->next = t2;
        p = p->next;
        l = t2;
    }
    else{
        pre = l;
        l = l->next;
    }
}
return fakeHead->next;
}
};

```