

链表分割

现有一链表的头指针 `ListNode* pHead`, 给一定值 `x`, 编写一段代码将所有小于 `x` 的结点排在其余结点之前, 且不能改变原来的数据顺序, 返回重新排列后的链表的头指针。

```
/*
struct ListNode {
    int val;
    struct ListNode *next;
    ListNode(int x) : val(x), next(NULL) {}
};*/
class Partition {
public:
    ListNode* partition(ListNode* pHead, int x) {
        // write code here
        if(pHead==NULL)
            return NULL;
        //构造两个节点头分别存放比 x 小的 和比 x 大的 最后大的接在小的后面
        //两个分别作为大小头结点 固定不一定
        ListNode* smallhead=(ListNode*)malloc(sizeof(ListNode));
        ListNode* bighead=(ListNode*)malloc(sizeof(ListNode));
        //同时给两个移动指针遍历
        ListNode* small=smallhead;
        ListNode* big=bighead;
        while(pHead)
        {
            if(pHead->val<x)
            {
                small->next=pHead;
                small=small->next;
            }
            else
            {
                big->next=pHead;
                big=big->next;
            }
            pHead=pHead->next;
        }
        big->next=NULL;
        //将 smaller 链表最后一个节点的指针域放置 bigger 链表的头指针
        //注意是 bighead 而不是 big , big 已经移动到末尾并置空
        small->next=bighead->next;
        return smallhead->next;
    }
};
```