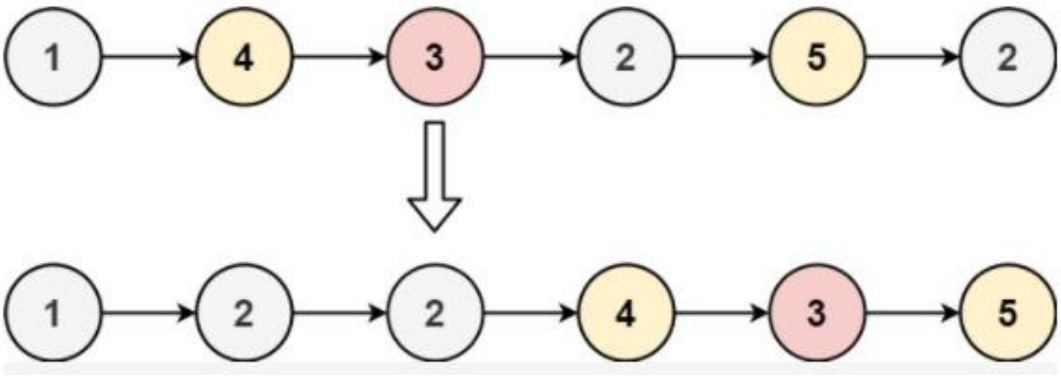
分隔链表

给你一个链表的头节点 `head` 和一个特定值 `x`，请你对链表进行分隔，使得所有 **小于** `x` 的节点都出现在 **大于或等于** `x` 的节点之前。

你应当 **保留** 两个分区中每个节点的初始相对位置。

示例 1：



输入： `head = [1,4,3,2,5,2], x = 3`

输出： `[1,2,2,4,3,5]`

示例 2：

输入： `head = [2,1], x = 2`

输出： `[1,2]`

```
/**
 * 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* partition(ListNode* head, int x) {
        ListNode* head1=new ListNode(0);
        ListNode* head2=new ListNode(0);
        ListNode* cur1=head1;
```

```
        ListNode* cur2=head2;
        while(head)
        {
            if(head->val<x)//小于 x 的存放 cur1
            {
                cur1->next=head;
                cur1=cur1->next;
            }
            else//大于 x 的存放 cur2
            {
                cur2->next=head;
                cur2=cur2->next;
            }
            head=head->next;
        }
        //合并连接
        cur1->next=head2->next;
        cur2->next=nullptr;
        return head1->next;
    }
};
```