

给定一个链表和一个特定值 x ，对链表进行分隔，使得所有小于 x 的节点都在大于或等于 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:

输入: head = 1->4->3->2->5->2, $x = 3$

输出: 1->2->2->4->3->5

分析，直接遍历树，将大于小于的放在左右链表上，到最后将两链表拼接返回即可。

```
/**
 * 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) {
        ListNode* right=NULL;
        ListNode* left=NULL;
        ListNode* right1=NULL;
        ListNode* left1=NULL;
        while(head!=NULL){
            if(head->val<x){
                if(left!=NULL){
                    left1->next=head;
                    left1=left1->next;
                }else{
                    left=head;
                    left1=head;
                }
            }else{
                if(right!=NULL){
                    right1->next=head;
                    right1=right1->next;
                }else{
                    right=head;
                    right1=head;
                }
            }
            head=head->next;
        }
        if(left!=NULL){
```

```
        left1->next=right;
        if(right1!=NULL){
            right1->next=NULL;
        }
    }else{
        left=right;
    }
    return left;
}
};
```