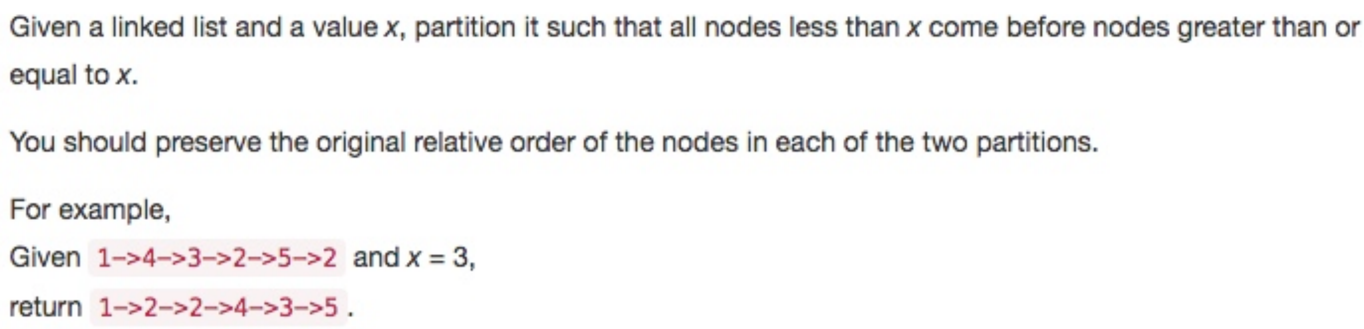
086. Partition List

086 Partition List

- **Two Pointers** + Linked List

Description



Given a linked list and a value *x*, partition it such that all nodes less than *x* come before nodes greater than or equal to *x*.

You should preserve the original relative order of the nodes in each of the two partitions.

For example,
Given `1->4->3->2->5->2` and *x* = 3,
return `1->2->2->4->3->5`.

1. Thought line

2. Two Pointers + Linked List

```
/**
 * 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* dummyHeadFirstHalf = new ListNode(0);
        ListNode* dummyHeadSecondHalf = new ListNode(0);
        ListNode* firstHalfElement = dummyHeadFirstHalf;
        ListNode* lastHalfElement = dummyHeadSecondHalf;

        while(head!=nullptr){
            ListNode* headNext = head->next;
            if(head->val<x){
                firstHalfElement->next = new ListNode(head->val);
                firstHalfElement = firstHalfElement->next;
            }
            else{
                lastHalfElement->next = new ListNode(head->val);
                lastHalfElement = lastHalfElement->next;
            }
            head = headNext;
        }
        firstHalfElement->next = dummyHeadSecondHalf->next;
        return dummyHeadFirstHalf->next;
    }
};
```