**Partition List**

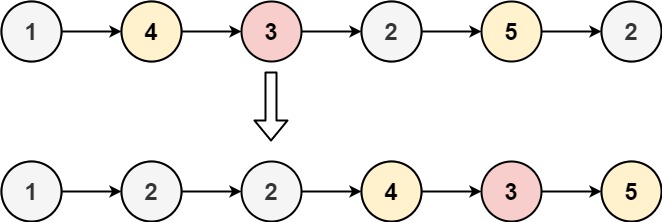
Medium

4143538Add to ListShare

Given the head of 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.

**Example 1:**



**Input:** head = [1,4,3,2,5,2], x = 3

**Output:** [1,2,2,4,3,5]

Code

class Solution {

public:

ListNode\* partition(ListNode\* head, int x) {

ListNode\* small= new ListNode(0);

ListNode\* higher = new ListNode (0);

ListNode \* smallHead=small;

ListNode \* highHead=higher;

while(head!=NULL)

{

if(head->val<x)

{

smallHead->next = head;

smallHead=smallHead->next;

}

else

{

highHead ->next = head;

highHead = highHead ->next;

}

head =head->next;

}

highHead->next =nullptr;

smallHead->next =higher->next;

return small->next;

}