86. Partition List

Add to List

Question | Editorial Solution | My Submissions (/problems/partition-list/submissions/)

Total Accepted: **87439** Total Submissions: **276867** Difficulty: **Medium** Contributors: **Admin**

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` .

Subscribe (/subscribe/) to see which companies asked this question

Show Tags

Have you met this question in a real interview? Yes No

Discuss (https://discuss.leetcode.com/category/94) | Top Solutions

Pick One (/problems/random-one-question/)

C++

```
/**
 * 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) {
        if(head==NULL) return NULL;

        ListNode* greater = new ListNode(0);
        ListNode* dummy = new  ListNode(0);
        dummy->next = head;
        ListNode* cur = dummy;
        ListNode* gt = greater;
        while(cur->next!=NULL){
            if(cur->next->val < x) cur = cur->next;
            else {
                gt->next = cur->next;
                cur->next  = cur->next->next;
                gt = gt->next;
            }
        }
        cur->next = greater->next;
        gt->next = NULL;
        return dummy->next;
    }
};
```

Notes

Custom Testcase

Contribute Testcase

Run Code | Submit Solution