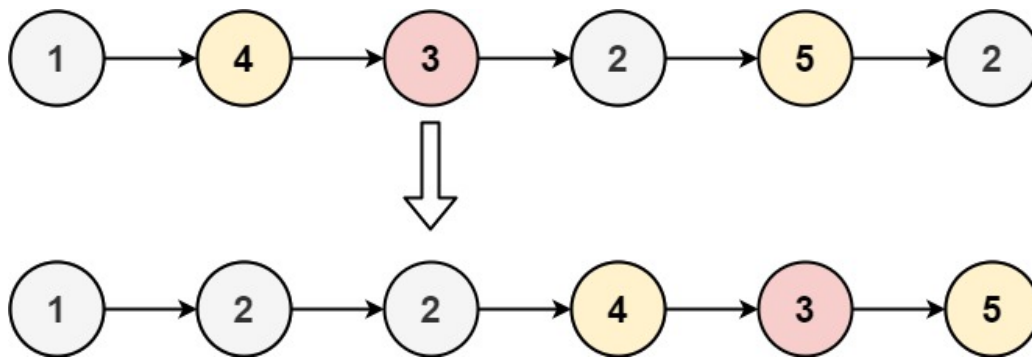


[Description](#)
[Solution](#)
[Discuss \(999+\)](#)
[Submissions](#)
i C++

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]

Example 2:

Input: head = [2,1], x = 2

Output: [1,2]

Constraints:

- The number of nodes in the list is in the range $[0, 200]$.
- $-100 \leq \text{Node.val} \leq 100$
- $-200 \leq x \leq 200$

Accepted 381,871

Submissions 766,205

Seen this question in a real interview before?

Companies

Related Topics

Similar Questions

```

9      * };
10     */
11     class Solution {
12     public:
13         ListNode* partition(ListNode* head, int x) {
14             if (!head) return head;
15             ListNode* dummyOne = new ListNode(0);
16             ListNode* dummyTwo = new ListNode(0);
17             dummyOne->next = dummyTwo;
18             ListNode* curr = head;
19             while (curr) {
20                 if (curr->val < x) {
21                     dummyOne->next = curr;
22                     dummyOne = curr;
23                 } else {
24                     dummyTwo->next = curr;
25                     dummyTwo = curr;
26                 }
27                 curr = curr->next;
28             }
29             dummyOne->next = dummyTwo->next;
30             return dummyOne;
31         }
    
```

Your previous code was:

Testcase

Run Code Re

Accepted

Runti

Your input

[1,
3]

Output

[1,

Expected

[1,

Console

Use

Run Code

Problems

Pick One

< Prev

86/2345

Next >