

# o86. Partition List

## o86 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  $1 \rightarrow 4 \rightarrow 3 \rightarrow 2 \rightarrow 5 \rightarrow 2$  and  $x = 3$ ,

return  $1 \rightarrow 2 \rightarrow 2 \rightarrow 4 \rightarrow 3 \rightarrow 5$ .

### 1. Thought line

### 2. Two Pointers + Linked List

```
1 /**
2  * Definition for singly-linked list.
3  * struct ListNode {
4  *     int val;
5  *     ListNode *next;
6  *     ListNode(int x) : val(x), next(NULL) {}
7  * };
8  */
9 class Solution {
10 public:
11     ListNode* partition(ListNode* head, int x) {
12         ListNode* dummyHeadFirstHalf = new ListNode(0);
13         ListNode* dummyHeadSecondHalf = new ListNode(0);
14         ListNode* firstHalfElement = dummyHeadFirstHalf;
15         ListNode* lastHalfElement = dummyHeadSecondHalf;
16
17         while(head != nullptr){
18             ListNode* headNext = head->next;
19             if(head->val < x){
20                 firstHalfElement->next = new ListNode(head->val);
21                 firstHalfElement = firstHalfElement->next;
22             }
23             else{
24                 lastHalfElement->next = new ListNode(head->val);
25                 lastHalfElement = lastHalfElement->next;
26             }
27             head = headNext;
28         }
29         firstHalfElement->next = dummyHeadSecondHalf->next;
30         return dummyHeadFirstHalf->next;
31     }
32 };
```