

LeetCode

Explore

Problems

Interview

Contest

Discuss

Store

Description

Solution

Discuss (999+)

Submissions


876. Middle of the Linked List

Easy 6430 168 Add to List Share

Given the `head` of a singly linked list, return the *middle node of the linked list*.

If there are two middle nodes, return the **second middle** node.

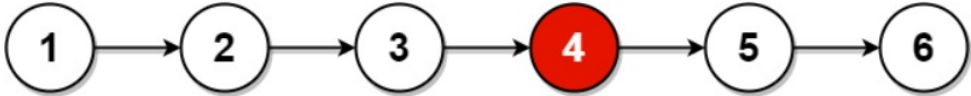
Example 1:



```
graph LR; 1((1)) --> 2((2)); 2 --> 3((3)); 3 --> 4((4)); 4 --> 5((5));
```

Input: `head = [1,2,3,4,5]`
Output: `[3,4,5]`
Explanation: The middle node of the list is node 3.

Example 2:



```
graph LR; 1((1)) --> 2((2)); 2 --> 3((3)); 3 --> 4((4)); 4 --> 5((5)); 5 --> 6((6));
```

Input: `head = [1,2,3,4,5,6]`
Output: `[4,5,6]`
Explanation: Since the list has two middle nodes with values 3 and 4, we return the second one.

Java

Autocomplete

```
1 /**
2  * Definition for singly-linked list.
3  * public class ListNode {
4  *     int val;
5  *     ListNode next;
6  *     ListNode() {}
7  *     ListNode(int val) { this.val = val; }
8  *     ListNode(int val, ListNode next) { this.val = val; this.next = next; }
9  * }
10 */
11 class Solution {
12     public ListNode middleNode(ListNode head) {
13         ListNode slow=head;
14         ListNode fast=head;
15
16         while(fast!=null && fast.next!=null){
17             slow=slow.next;
18             fast=fast.next.next;
19         }
20
21         return slow;
22     }
23 }
24
```

Problems

Pick One

Prev 876/2371 Next

Console Contribute

Run Code

Submit

29°C Cloudy

ENG IN 12:08 PM 11-08-2022