链滴

# [每日 LeetCode] 876. Middle of the Linked List

作者：Hanseltu
原文链接：https://ld246.com/article/1555604692943
来源网站：链滴

Description:

Given a non-empty, singly linked list with head node head, return a middle node of linked list.

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

**Example 1:**

Input: [1,2,3,4,5]
Output: Node 3 from this list (Serialization: [3,4,5])
The returned node has value 3. (The judge's serialization of this node is [3,4,5]).
Note that we returned a ListNode object ans, such that:
ans.val = 3, ans.next.val = 4, ans.next.next.val = 5, and ans.next.next.next = NULL.

**Example 2:**

Input: [1,2,3,4,5,6]
Output: Node 4 from this list (Serialization: [4,5,6])
Since the list has two middle nodes with values 3 and 4, we return the second one.

**Note:**

- The number of nodes in the given list will be between 1 and 100.

---

思路：本题要求找单链表的中点。考虑使用快慢指针，慢指针走一步，快指针走两步，快指针走到末时，慢指针即指向中间节点。

---

C++代码

```
/**
 * Definition for singly-linked list.
 * struct ListNode {
 *     int val;
 *     ListNode *next;
 *     ListNode(int x) : val(x), next(NULL) {}
 * };
 */
class Solution {
public:
    ListNode* middleNode(ListNode* head) {
        ListNode* slower = head;
        ListNode* faster = head;
        while (slower && faster && faster->next)
        {
            slower = slower->next;
            faster = faster->next->next;
        }

        return slower;
    }
```

```
};
```

---

运行时间：4ms

运行内存：8.3M