LintCode 参考程序

Lionx

2016 年 4 月 19 日

目录

第一章　入门（Naive）

1.1 Problem ID: 228 Middle of Linked List

1.1.1 Description

Find the middle node of a linked list.

1.1.2 Example

Given 1->2->3, return the node with value 2.

Given 1->2, return the node with value 1.

1.1.3 Code

C++

```
/**
 * Definition of ListNode
 * class ListNode {
 * public:
 *     int val;
 *     ListNode *next;
 *     ListNode(int val) {
 *         this->val = val;
 *         this->next = NULL;
 *     }
 * }
 */
class Solution{
public:
    /**
     * @param head: the head of linked list.
     * @return: a middle node of the linked list
     */
    ListNode *middleNode(ListNode *head) {
        // Write your code here
        if(head == NULL){
            return NULL;
        }
        ListNode *fast = head;
        ListNode *slow = head;
        while(fast->next != NULL && fast->next->next != NULL){
            slow = slow->next;
            fast = fast->next->next;
        }
        return slow;
    }
};
```

Python

```python
"""
Definition of ListNode
class ListNode(object):

    def __init__(self, val, next=None):
        self.val = val
        self.next = next
"""

class Solution:
    # @param head: the head of linked list.
    # @return: a middle node of the linked list
    def middleNode(self, head):
        # Write your code here
        if head is None:
            return None
        slow = head;
        fast = head;
        while fast.next is not None and fast.next.next is not None:
            slow = slow.next
            fast = fast.next.next
        return slow
```

第二章　容易（Easy）

第三章　中等（Medium）

第四章　困难（Hard）

第五章　超难（Super）