92. Reverse Linked List II

Description | Hints | Submissions | Discuss | Solution

Pick One

Reverse a linked list from position *m* to *n*. Do it in one-pass.

Note: 1 ≤ *m* ≤ *n* ≤ length of list.

Example:

```
Input: 1->2->3->4->5->NULL, m = 2, n = 4
Output: 1->4->3->2->5->NULL
```

Seen this question in a real interview before? Yes No

```
public class ListNode{
    int val;
    ListNode next;
    public ListNode(int val) {
        this.val = val;
    }
}
 public ListNode reverseBetween(ListNode head, int m, int n) {
     if (m == n) {
        return head;
    }

     ListNode preHead = new ListNode(0);
     preHead.next = head;
     //找到第m个节点
     ListNode firstTail = preHead;
     int k = m - 1;
     while (k -- > 0) {
        firstTail = firstTail.next;
    }

     ListNode secondTail = firstTail.next;

     ListNode tmpHead = null;
     ListNode tmpNext = null;
     ListNode node = firstTail.next;
     k = n - m + 1;
     while (k -- > 0) {//第n个节点和第m个节点之间的反转
        tmpHead = node;
        node = node.next;
        tmpHead.next = tmpNext;
        tmpNext = tmpHead;
    }
     //将三部分连接
     firstTail.next = tmpHead;
     secondTail.next = node;

     return preHead.next;
 }
}
```