给定一个列表，将列表向右旋转*k个*地方，其中*k*是非负数。

**例：**

给定**1-> 2-> 3-> 4-> 5-> NULL**并且*k* = **2**，

返回**4-> 5-> 1-> 2-> 3-> NULL**。

题目已经很明确

我们需要找到倒数第k个节点，然后把倒数第k个节点变成头指针，然后把头指针放到最后一位。这是在k不大于链表长度的情况。

我们这里采用双指针的方法，使用慢指针q指到倒数第k+1个节点，即倒数第k个节点的前一个节点。同时，先行指针p走到最后一个节点。正常的情况下，应该是先行指针走到最后一个指针的下一个节点，即最后一个节点的.next()，q指针到倒数第k节点。

我们找到了倒数第k+1的节点，然后进行交换还有断链，不断的话就成为循环单链表了与题目不符合。

节点交换

1.p.next = head; //head节点后移，最后一个节点的next指向头结点

2.head = q.next;//倒数第k个节点前移，head变成q.next节点

3.q.next = null;//断链，在倒数第k+1个节点处断链

时间复杂度O(n)

我们还需要考虑一种情况就是k很大的时候，k已经长于链表的本身了。当遇到这种情况，我们不得不先记录链表得长度length，然后用k%length得出要移动得是哪个节点，即k，这样做得时间复杂度就会增加为O(2\*n)，但是量级没有发生变化依然是O(n).如上述方法，我们先进行对链表的计数，然后处理k，最后再查找k进行交换。

代码