

单链表面试集合

找出单聊表中倒数第K个的元素

使用指针追赶的方法，定义两个指针fast和slow，fast先走K步("注意链表长度是否大于K")，然后fast和slow在同时继续走，当fast到链表尾部时，slow指向倒数第K个。

找出单链表的中间元素

使用指针追赶的方法，fast每次走两步，slow每次走一步，当fast到链表尾部时，slow指向链表中间元素。

判断单链表是否有环？

使用指针追赶的方法，shlow每次都一步，fast每次都两步、如果存在则两者相遇；如果不存在则fast遇到Null退出

如何知道环的长度

记住一个点，让slow从这里出发，环着走一圈，再次回到点时，所经过的节点数就是环的长度

判断两个链表(无环)是否相交

- 遍历两个链表，进行比较看节点是否相同
- 两链表一旦相交，相交节点一定有相同的内存地址。遍历第一个链表，并利用地址建立哈希表，遍历第二个链表，看看地址哈希值是否和第一个表中的节点地址值有相同即可判断两个链表是否相交。时间复杂度 $O(\text{length}(A) + \text{length}(B))$
- 两个链表相连接变成一个链表，slow查看是否有环

判断两个链表(可能有环)是否相交

- 都无环 同上
- 一个有环 一个没有 不相交
- 两个都有，在A链表上，使用指针追赶的方法，找到两个指针碰撞点，之后判断碰撞点是否在B链表上。如果在，则相交。

寻找两个相交链表的第一个公共节点

- 双重for循环进行查找，直到找到相等的节点， $O(mn)$

- 相交的点在环外。如果两个链表有公共节点，那么该公共节点之后的所有节点都是两个链表所共有的，所以长度一定也是相等的，如果两个链表的总长度是相等的，那么我们对两个链表进行遍历，则一定同时到达第一个公共节点。如果链表的长度实际上不一定相同，所以计算出两个链表的长度之差 n ，然后让长的那个链表先移动 n 步，短的链表再开始向后移动的时候比较两个链表的节点是否相等就可以获得第一个公共节点。时间复杂度是 $O(m+n)$