

Assignment 1 Report

单链表反转问题

曹旭 大数据学院 16307110230

关于单链表反转我想到的就有三种解题思路，下文将主要论述我选的解题方法的对应思路，同时简述其它两种方法的思路，所有方法我都在自己的电脑上运行过了。

方法一：新建链表，头节点插入法

我最开始上传测试的就是这个方法，其思路为：新建一个链表的头结点，遍历原链表，把每个节点头插入到新建链表中。最后，新建的链表就是反转后的链表。该方法充分利用了头插法的便利性，直接调用我提前写好的头插法 `HeadInsert` 函数就轻松解决了问题。这一思路的时间复杂度为 $O(n)$ ，空间复杂度为 $O(n)$ 。方法优点：调用现成的函数，节省了码代码的时间；缺点：空间复杂度过高，实际上可以直接在原链表上操作，如方法二。

方法二：直接（就地）反转法

新建一个指针遍历链表，并把当前指向的下一个节点 `pCur` 从原位置取出并插入到头结点 `L` 的下一个节点中，就地反转。

如下：

`L->1->3->5->7->9` 的就地反转过程：

`L->3->1->5->7->9`

`L->5->3->1->7->9`

`L->7->5->3->1->9`

`L->9->7->5->3->1`

该方法时间复杂度为 $O(n)$ ，空间复杂度为 $O(1)$ 。

方法三：使用堆栈

这个方法也很好理解，依次 `push` 进栈，然后再全部 `pop` 出，顺序正好和原链表相反。时间复杂度为 $O(n)$ ，空间复杂度为 $O(n)$ 。

个人在测试后感觉方法一、三很方便实现，而方法二在空间复杂度上优于方法一、三。