

第一次编程作业Report

罗华坤 软件02 2019011799

第一题

实现方法

利用循环列表的方式实现逆时针行走的特性，每个 *ListNode* 都逆时针地指向下一节点。

通过写一个向右插入的函数来实现面试者坐下的功能。

对列表进行 $(m - 1) \% n$ 次遍历（ n 为当前的面试者数），找到当前面试者的位置，减少不必要的遍历次数。

最后再用 *dfs* 进行逆序输出，也可采用循环列表的方式顺时针遍历。

遇到的问题

没注意到经过第一个人也算1位，因此遍历次数变为了 $m \% n$ （ n 为当前的面试者数）

时间、空间复杂度

时间复杂度为一共有 n 位面试者，每次找座位最坏都需要 $O(n)$ 的时间，因此最坏的时间复杂度为 $O(n^2)$

空间复杂度即为 $O(n)$

第二题

实现方法

由于中转盲端有后进先出（*LIFO*）的特性，因此采用栈的数据结构予以实现。

利用一个 *int current* 来指示当前未入栈的车序号的顶端（也可用一个栈实现）

每个大循环内，先不断判断即将入栈的元素（*current*）是否小于等于出栈序列的栈顶（*target[ans]*）

是则入栈，*current* ++，否则继续大循环内的操作，即不断判断能否出栈。

能否出栈仅需判断栈顶是否与出栈序列的栈顶相等，一旦相等，出栈，且 *ans* ++。

直至 *ans* 到达 n ，说明已比对完全；或入栈出栈循环都未进入，说明序列不可能产生。

遇到的问题

不知道何时判否，思考后发现，只要一个大循环内未执行入栈或出栈操作便可认为无法匹配。

时间、空间复杂度

每个元素都最多入栈与出栈一次，因此时间复杂度为 $O(n)$ 。

空间上用了个栈与两个数组储存操作与期望序列，因此空间复杂度为 $O(m + n)$ 。

第三题

解法与遇到的问题

从邓俊辉老师的 *mooc* 中得到提示 <https://www.bilibili.com/video/BV1hj411S7wU?p=102>

由于暴力遍历的方法无法完全通过 $O(n^2)$ ，因此采用求逆序对的方式实现。

而该题的输入是无序且包含两个值的点，因此我们需要先对 x 进行排序。

再对 y 坐标求逆序对，也就是本题的输出（采用 *long long int*）。

为了与求逆序对对应，均采用归并排序实现 $O(n \log n)$

而利用归并排序的特性，逆序对可以在归并排序中加一句话实现。

时间空间复杂度

用于采用两次归并排序，因此时间复杂度为 $O(n \log n)$ ，空间复杂度为 $O(n)$