



FOI2023 算法夏令营B班

第二天讲评

南方科技大学 匡亮



今天的奖品

- 第一名：明信片+咖啡杯
- 第二名：搪瓷杯
- 不在场就顺延
- 不止两个人AK就随机数生成器



后缀表达式求值

- 考点：栈
- 出题人生怕你没看出来这题在问的就是后缀表达式，直接把答案写在了标题上
- 从左往右枚举字符串内的元素，维护一个栈，遇到数字就入栈，遇到符号就出栈两个数字进行运算然后入栈
- 如果做到最后栈内恰好剩一个元素，第一行输出1第二行输出这个元素，否则两行都输出0
- 时间复杂度 $O(n)$



矩阵字典序

- 考点：扩展域并查集
- $A[i][j]$ 和 $A[j][i]$ 在最后的矩阵中会被交换当且仅当 i 和 j 恰好有一个被操作过
- 按照重要程度顺序，从左上角往右下角枚举（只枚举主对角线上方的元素即可，当然只枚举下方或者全部枚举也行）
- 如果这个位置必须交换，那么 i 和 j 的状态必须不同；如果必须不交换，那么 i 和 j 的状态必须相同
- 如果新的条件和之前条件矛盾就忽略
- 本质上来说是一个2-SAT问题，时间复杂度 $O(n^2)$



才羽绿的难题

- 考点：单调队列（更好的做法）或单调栈
- 先讲比较不优美的做法
- 如果在某个位置发生了最大值更新，下一次发生最大值更新的一定是后面一个固定的位置
- 这个信息可以用单调栈处理出来
- 维护一个栈，里面是“还没找到下一个比自己大”的元素，那么它们是从大到小的（否则栈靠上的元素就是栈靠下的元素的“下一个比自己大”的元素）
- 新元素把栈顶比自己小的元素的“下一大”设为自己然后弹出，自己放在栈顶即可



才羽绿的难题

- 处理出这个信息以后，不难发现这个跳转关系是一个森林（添加一个虚根变成一棵树也行）
- 由于每个区间的第一个位置一定发生最大值更新，从这个位置开始树上倍增就好了，时间复杂度 $O(n\log n)$
- 这种做法比较直观，虽然不够巧妙，但今天还是放过了
- 当然还有一种更好的做法



才羽绿的难题

- 更好的做法用到了正难则反的思想
- 假设我们用单调队列来从左往右做这个问题，每次删掉队首的元素后，下一个加入的元素可能会带来不止一个更新位置，非常烦人
- 但反之，如果我们从右往左做这个问题，每次删掉队首（现在是最右边）的元素则没有任何影响，而在队尾（最左边）插入新的元素时，更新位置的数量只会变少
- 因此用单调队列解决即可，时间复杂度 $O(n)$
- 这题本身难度不大，但可以同时用单调栈和单调队列解决，并且用到了我很喜欢的正难则反思想，因此放在最后一题



谢谢大家!

