FOI2023算法夏令营B班
第三天讲评

南方科技大学 匡亮

今天的奖品

- 第一名： 明信片+星空杯
- 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

玩具装箱

- 知识点：线段树
- 考虑平方暴力：对于一些已知的玩具和箱子，我们如何求出答案。显然我们要将物品从小到大排序，相同大小则玩具在前。之后我们从小到大枚举：
  - 枚举到玩具的时候，事实上它和之前所有出现过的玩具是等价的，因为之后即将出现的箱子一定可以装下它们中的任何一个。
  - 枚举到箱子的时候，它一定会马上装入一个现有的玩具（如果存在），因为之后出现的玩具只会更大。如果不存在玩具，直接丢弃。
- 不难发现这其实就是括号匹配。于是我们离线存下所有物品，然后离散化用普通线段树解决即可。时间复杂度O(NlogN)。

小朋友的mex

- 知识点：线段树
- 既然是数据结构题，啥也不想，先打个平方暴力。
- 先枚举左端点，然后枚举右端点，我们希望能快速更新出答案。
- 我们发现，我们可以用一个bool数组来记录当前所有出现过的数，然后用一个数记一下当前的mex。
- 任何时候，mex+1的数如果出现了，我们就不断++mex，直到mex+1的数是没有出现的。
- 在一次暴力中，++mex的次数不会超过n次，所以复杂度是$O(n^2)$的。

小朋友的mex

- 接下来重要的部分分，很可惜事实上它对正解的启发意义不大
- 对于一个排列，由于每个数都是唯一的，所以你想让mex至少是x的时候，当且仅当你包含了0到x-1所有数
- 因此维护前x个元素中的最靠左和最靠右的位置分别是啥，然后把它们到边界的距离乘起来（表示有多少个区间可以包含他们）就好了

小朋友的mex

- 这类问题的常见思路是枚举右端点后，考虑如何一次性算出所有以这个右端点结尾的区间的答案的和。
- 计算mex和数颜色很类似，如果我们枚举了右端点，我们就只关心每个数最后一次出现的位置。
- 假设我们用某种数据结构维护了当前右端点对应的所有左端点的答案，并假设我们新加入的右端点是0，那么：
- Ai： 1 0 2 3 1
- 后缀mex： 4 4 0 0 0
- Ai： 1 0 2 3 1 0
- 后缀mex： 4 4 4 2 2 1

小朋友的mex

- Ai：1 0 2 3 1
- 后缀mex：4 4 0 0 0
- Ai：1 0 2 3 1 0
- 后缀mex：4 4 4 2 2 1
- 我们发现事情变得有些麻烦，因为被更新的答案中出现了1、2、4三种。但如果我们反过来看，则是1、2、4三种数字都被更新成了0。因此我们想到，这题我们应该反过来做。
- 从右往左枚举右端点，每次计算出这个右端点的答案之和后，考虑维护删掉这个右端点之后所有左端点的答案会如何变化。

小朋友的mex

- 假设我们枚举到删掉的i这个右端点，设Ai的值为x，我们想办法找到上一次x出现的位置last，那么：
  - 位置≤last的所有左端点答案不会变（在它们看来，右边的数组成的集合没有发生变化）；
  - 位置在last到i之间的所有位置，如果原本的答案小于等于x，则不会受影响（在x之前，已经有一个自然数没有出现过了）；
  - 如果原本的答案大于x，则答案会变成x（x变成了最小的没有出现的数字）。
- 也就是说，last+1到i之间的答案要对x取min。
- 那么在删掉一个右端点之前，我们对某种数据结构上的所有点的答案求和，然后删掉这个右端点，再往左走一步即可。
- 也就是说，我们的数据结构得支持两种操作：区间对一个数取min，区间求和。

小朋友的mex

- 区间取min、区间求和是一个高级的线段树问题，但我们这里并不需要这么麻烦。
- 注意到我们的线段树中存的是一个（从左往右）单调递减的序列，我们只要（在线段树上）二分到对应的区间然后进行区间赋值就好了。
- 时间复杂度O(nlogn)。

兰登战术

- 知识点：二分答案、线段树分治、线段树上二分
- 拿到吓人的式子先别着急，看看有什么特点：
- 首先映入眼帘的是外层min内层max，这是经典的最大值最小化结构，基本锁定了答案中有一步“二分答案”
- 接着，操作只有插入和删除，而且要维护的是极值相关的信息，我们知道，删除对极值的影响非常大，所以基本锁定了答案中有一步“线段树分治”

兰登战术

- 用上了这两步，题目变成了：
- 只需要支持“加入”（和撤销）操作；如果是(A1,B1)加入S1，二分答案x，询问S2中是否有元素满足max(A1+A2,B1+B2)≤x，将二分后的x和加入(A1,B1)前的答案作对比，取更小的作为新的答案；如果是(A2,B2)加入S2要做的事情完全一样
- 由于我们已知A1,B1,x，max(A1+A2,B1+B2)≤x可以拆成A2≤x-A1且B2≤x-B1
- 我们把A看成每个元素的下标，B看成每个元素的值，整个问题变成了单点修改+查询前缀最小值，树状数组即可

兰登战术

- 这个做法的复杂度为$O(n\log^3 n)$，不过优化的思路也非常明确：把二分答案+树状数组改成在线段树上二分
- 假设我们直接在线段树上二分答案x，判断x≤mid能否成立，那么能选用的A2必须满足A2≤mid-A1，选出的B2必须满足B2≤mid-B1
- 显然B2就是区间[1,mid-A1]的最小值，我们就是在问[1,mid-A1]的最小值是否≤mid-B1
- 然而现在有一个问题，[1,mid-A1]的最小值没法O(1)得到

兰登战术

- 我们稍微转变一下，变成在线段树上二分(x-A1)的值最小是多少
- 那么我们需要判断的是x-A1≤mid能否成立，那么能选用的A2必须满足A2≤mid，对应的B2就是[1,mid]的最小值，这个是我们可以O(1)得到的：我们可以维护Fmin表示[1,L-1]的最小值，那么min(Fmin,T[Ls])就是[1,mid]的最小值（T表示线段树节点，Ls表示左孩子）；如果我们选择往左子树走，Fmin不变；如果我们选择往右子树走，Fmin就变成min(Fmin,T[Ls])
- 我们选出来的B2应该满足B2≤mid+A1-B1，其中A1-B1是一个常数，直接传进来就好了
- 时间复杂度$O(n\log^2 n)$

Southern University of Science and Technology

谢谢大家！