

# NOIP模拟赛题解

---

## T1

---

注意到一个重要的性质：修改不会出现一个完全覆盖另一个的情况。

注意到另一个重要的性质：对于一个已经确定的前缀，考虑对之后的影响只有三种状态，要么它处于一个修改的中的某一位，要么这个位置未经修改，要么处于两种都有可能状态。

容易发现这样只维护当前位置是否修改是正确的，因为倒数第二靠后的修改不可能对之后产生任何影响。

对于每一种类型的前缀维护数量即可，复杂度  $O(nm)$ 。

## T2

---

考虑先贪心吃掉所有小于自己初始值的数，然后我们发现如果还能吃掉任何一个比自己初始值大的数，那么自己能否获胜与那个数能否获胜等价。

因此我们需要维护周围小于自己的数的sum和这个连通块相邻的最小的数（因为最小的也能等价，而且最小的是最可能被吃掉的）。

这显然可以用线段树合并维护。复杂度  $O(nm \log^2 nm)$

其实利用这个性质可以考虑暴力吃掉所有能吃的，中间只要碰到任意一个比自己初始值大且能吃掉的数就停下。如果一个数不能吃掉，就不用访问自己访问过的点。实测在随机数据下跑得很快，面向数据进行特殊优化应该能通过此题。复杂度未知

## T3

---

以竖着删为默认代价，考虑横着删的价值。容易发现同一竖从下往上价值递减。

更进一步，越靠左减得越快。因此可以当作直线处理，求上凹包。显然上凹包都是可以取到的。

使用主席树维护，复杂度  $O(n \log n)$ 。

std没有线段树上二分，因此  $O(n \log^2 n)$ 。我懒得写子

## T4

---

首先考虑只划分成若干个大小固定的块。

首先求出一个特解，然后容易发现将特殊点移动只会产生  $O(1)$  的影响。因此可以维护。

然后同时使用树哈希维护树的形态。

代码量略大，复杂度  $O(n \log n)$ 。