

• 贪心

- 题外话

- czc 反映有同学觉得昨天的题目太简单，于是这里给学有余力的同学留俩题：
- 「CF875E」 Delivery Club
- 「AGC032E」 Modulo Pairing

- 贪心

- 贪心是一个更神秘的东西。
- 无非就是你自己要找到一种最优秀的策略。
- 所以今天我们直接来做题。
- 上午讲的题目难度相对较低，留给大家充足的时间消化昨天的题目（不是

• 「UOJ 136」开学前的作文

•  $n, m \leq 10^9$ 。

红包是一个萌萌的男孩子。

红包由于 NOI 惨挂，直到前不久依然无心写作业。如今快开学了，他决定好好完成作业。

对于可以交电子稿的作文，红包有特殊的完成技巧，大致流程是依次选中一段内容→按下 **Ctrl+C** →按下 **Ctrl+V**。

由于红包的键盘过于奇特，只有 **↑**，**↓**，**←**，**→**，**Fn** 这 5 个按键能够移动光标，导致每次选中一段内容总要费上一番功夫。

按 键	功能
<b>↑</b>	将光标向上移动一格
<b>↓</b>	将光标向下移动一格
<b>←</b>	将光标向左移动一格
<b>→</b>	将光标向右移动一格
<b>Fn</b>	依次按下最近被物理按下的两个 <b>↑</b> ， <b>↓</b> ， <b>←</b> ， <b>→</b> 键 例子：若依次按下 <b>↓</b> ， <b>Fn</b> ， <b>←</b> ， <b>Fn</b> ，则第一次按下 <b>Fn</b> 后等价于按下 <b>↓</b> ，第二次按下 <b>Fn</b> 后等价于依次按下 <b>↓</b> ， <b>←</b> 。

现在红包想要在按键次数尽可能少的情况下移动光标到第  $n$  行第  $m$  列。为了简化问题我们默认光标起始位置为第一行第一列。

- 「UOJ 136」开学前的作文

- 考虑最优策略是什么：
- 我们希望尽可能多的操作可以被  $F_n$  代替。
- 那么不妨假设  $n \leq m$ , 一种显然的策略是  
 $\rightarrow \downarrow F_n F_n \cdots F_n \rightarrow F_n F_n \cdots F_n (\rightarrow)$
- 判断即可。

- 「CF521D」 Shop

- 给定一个长度为  $n$  的正整数数组  $a$ 。
- 你现在有  $m$  个已经给定的操作  $(t, i, b)$ :
  - $t = 1$ , 则将  $a_i$  赋值为  $b$ 。
  - $t = 2$ , 则将  $a_i$  加上  $b$ 。
  - $t = 3$ , 则将  $a_i$  乘以  $b$ 。
- 你可以从中选出至多  $k$  个操作, 并按一定顺序执行。
- 请最大化  $\prod_{i=1}^n a_i$  的值, 并求出方案。

- 「CF521D」 Shop

- 如果只有第三种操作？
- 直接做就完了。

- 「CF521D」 Shop

- 如果只有第二种操作？
- 直接做就完了。



- 「CF521D」 Shop

- 如果只有第二、三种操作？
- 注意到如果我们要使用加法操作，对于同一个数字加法操作的使用顺序一定是固定的——即从大到小。
- 那么如果加法操作顺序固定，我们显然可以将其转化为乘法，贪心即可。

- 「CF521D」 Shop

- 如果存在一操作？
- 注意到一操作可以转化为加法，进而可以像刚才的情况一样做。
- 注意到在实际操作时赋值操作一定排在第一位，那为什么我们可以这样做呢？
- 全部转化为乘法后，由于乘法具有交换律，所以如果关于赋值操作转化为的乘法操作，我们可以提前到第一个完成而没有影响。

- 「CF1380G」 Circular Dungeon

- 有  $n$  个排列在环上的房间，第  $i$  个房间只能到达第  $i + 1$  个房间（第  $n$  个房间只能到达第 1 个房间），同时有  $n$  个宝箱，第  $i$  个宝箱的价值为  $c_i$ ，其中恰好有  $k$  个宝箱为假宝箱，每个房间有且仅有一个宝箱。
- 玩家等概率从  $n$  个房间选取一个房间开始移动，若当前房间有真宝箱，他将获得此宝箱的价值，否则立刻结束游戏。最后获得的总收益等于之前收集测宝箱的总价值。
- 对于  $k \in [1, n]$ ，你可以决定宝箱的放置位置和真假。
- 请求出玩家收益的最小期望值。
- $n \leq 3 \times 10^5$ 。

- 「CF1380G」 Circular Dungeon

- 先考虑  $k$  固定时的做法。
- 有一个显然的结论是，你一定会将价值前  $k$  大的宝箱全部变为假宝箱。
- 那么接下来呢？不妨考虑对于一段连续的真宝箱，其期望价值为多少。
- 根据这一结果容易知道越大价值的宝箱应当越靠前，且这一段的长度应尽可能短。

- 「CF1380G」 Circular Dungeon

- 在这个问题解决后，问题可以抽象为：现在有  $k$  个假宝箱围成一个环，你要向其中插入若干个真宝箱，使得最后玩家获得价值最小。
- 似乎问题已经解决了。

- 「CF505E」 Mr. Kitayuta vs. Bamboos

- 给定  $n$  个数  $h_{1..n}$ 。
- 你需要进行  $m$  轮操作, 每轮操作为  $k$  次修改, 每次修改可以选择一个数  $h_i$  修改为  $\max(h_i - p, 0)$ 。
- 每轮操作后每个  $h_i$  将会被修改为  $h_i + a_i$ 。
- 你需要最小化最终  $h_{1..n}$  中的最大值。
- $n \leq 10^5, m \leq 5 \times 10^3, k \leq 10$ 。

- 「CF505E」 Mr. Kitayuta vs. Bamboos

- 如果没有每轮只能进行  $k$  次修改的限制或者没有竹子长度必须大于 0 的限制那么直接贪心就完事了。
- 但是很遗憾。
- 首先看到最小化最大值可以想到用二分将最优化问题转化为判定性问题。
- 设当前二分的值为  $H$
- 但是有这个必须大于 0 的限制很烦，导致我们不一定能真的减少  $p$ 。

- 「CF505E」 Mr. Kitayuta vs. Bamboos

- 所以我们考虑让时光倒流。
- 即：假设所有的竹子最后都满足限制，即均为  $H$ 。每一次长高就相当于减少  $a_i$ ，每次你可以选择  $k$  根竹子使其长高  $p$ ，问最后竹子高度是否都  $\geq h_i$ 。
- 然后我们每一轮将最有可能掉为负数的那些竹子拉出来长高，用一个堆维护就完事了。



- 题单 1

- 「UOJ 136」 开学前的作文
- 「CF521D」 Shop
- 「CF1380G」 Circular Dungeon
- 「CF505E」 Mr. Kitayuta vs. Bamboos

- 题单 2

- 「CTSC2007」 数据备份
- 「洛谷 P5290」 春节十二响
- 「CSP-S2020」 贪吃蛇