

原本味道

俞涛正在清理大肠，他发现大肠中有一些原本味道，这些原本味道居然还在移动。

作为一名顶级厨师，俞涛打算保留大肠中的原本味道，于是他决定对这些原本味道进行一些研究。

经过长达 114514 秒的观察，他发现原本味道在大肠中的移动是有规律的：

大肠可以视为无限长，原本味道可以视为一条线段，初始时刻线段位于 $[0, 1]$ 。

原本味道的移动方式很有趣，具体来说它的移动可以视为左右端点的移动。

原本味道的右端点每 $a_1 + b_1$ 秒为一个循环，在前 a_1 秒每秒向右移动一个单位，后 b_1 秒静止不动；

原本味道的左端点每 $a_2 + b_2$ 秒为一个循环，在前 a_2 秒每秒向右移动一个单位，后 b_2 秒静止不动。

但俞涛并不满意，因为可能某个时刻原本味道的左端点会超过右端点，这样原本味道就消失了，违背了他保留大肠原本味道的初衷。

但是作为顶级厨师，俞涛很快找到了解决办法：如果原本味道的左端点向右移动时，右端点正在前方距离左端点恰好一个单位的位置静止，那么为了保留原本味道，左端点这一秒不会移动。

问题解决了，现在俞涛想知道经过 q 秒后原本味道的左端点移动了多少单位，这样他好截取附近的大肠来烹饪九转大肠。

由于俞涛要做很多道九转大肠，所以他会多次询问你不同的 q ，如果你能解决他的问题，那么他在给导师品尝九转大肠的时候，也会给你一份哦。

$$q \leq 10^{16}, a, b, n \leq 10^5。$$

首先：

$$Sr(t) = \left\lfloor \frac{t}{T_1} \right\rfloor \cdot a_1 + \min(a_1, t \bmod T_1) \quad (1)$$

我们争取求出所有的 del_t 表示 t 秒后两个人的距离，求出 $del_{T_1}, del_{2T_1}, \dots, del_{lcm(T_1, T_2)}$ 。若 $del_{T_1} = 1$ 且 $i \geq \frac{T_2}{\gcd(T_1, T_2)}$ ，那么其实就是循环了！（因为 $del_{iT_1 - lcm} = 1$ ）；否则就是发散了，因为左右端点已经完全独立了。

顶级厨师

俞涛想成为一名顶级厨师。

当然了，想成为一名顶级厨师，技能点肯定是不能少的。

具体的，俞涛打算在接下来的 n 天提升三种技能。

俞涛的精力有限，所以他一天只能提升一种技能；具体的，如果在第 i 天选择提升第 c 种技能，该技能会提升 $a_{i,c}$ 个等级。

同时如果他之前已经 k 天没有提升第 c 个技能，并且这天也没有提升他，那么他的这个技能等级会减小 k 。

具体的如果第 c 个技能上一次提升的时间是第 i 天，当前是第 j 天，并且这天提升的技能不是 c ，那么该技能等级会减少 $j - i$ 。

我们认为技能等级的最小值就是 0，不会出现负数。

同时，由于众所周知的原因，俞涛不关心某个特定技能等级的高低，他只希望三个技能等级的和尽可能大，所以他来找你，希望你帮他求出 n 天后他所有技能等级的最大值。

$$n \leq 1000, w \leq 10000。$$

首先一个技能不会荒废超过 \sqrt{w} ，否则可以调整；和 0 取 \max 就是选择一天开始计算一个人的贡献，其实一个人提早开始算贡献肯定是不优的，所以不用状压哪些人已经开始计算了。直接记下上一次是谁、剩下两个技能离多久，就能 dp 了。 $O(nw)$ 。

Colorful Tree Again

给定一棵有 n 个节点的树，边有边权和颜色。每个点有被摧毁和不被摧毁两种状态，初始所有点都没被摧毁。

一条简单路径指图中没有重复节点的路径。简单路径的长度定义为路径上所有边的边权之和。定义一条简单路径是好的，当且仅当路径仅有某一种颜色 c 构成，且所有颜色为 c 的边都在这条简单路径里，且路径上所有节点都没被摧毁。你需要处理两种操作：

1. 摧毁一个节点;
2. 修复一个节点。

每个操作之后,你都需要输出最长的好的路径长度。若没有输出 0。 $n \leq 2 \times 10^5$ 。

首先我们发现点分树做法显然,每个点处处理所有跨过它的链。

然后发现脑子有点问题啊,直接在每条链的 lca 处维护所有经过它的链的最大值就行了啊。

Codeforces Scoreboard

你在参加一场有 n 题的 CF 比赛。

解决每道题你都需要恰好 1 分钟,提交时间可忽略不计。任何时刻你都只能解决最多一道题。第 0 分钟比赛开始,故你可以在任何 $t \geq 1$ 时刻提交。假设你的提交一定通过。

每道题有三个参数 k_i, b_i 和 a_i , 表示在第 t 时刻解决该题会得到 $\max(a_i, b_i - k \cdot t)$ 的分数。

你需要合理安排解决题目的顺序以得到最高的分数。比赛时间可看作能够完成所有题目。 $n \leq 2 \times 10^5$ 。

首先肯定按照 k 从小到大选。设 $f_{i,j}$ 表示前 i 个选了 j 个的最大收益。那么这个关于 j 是凸的。

考虑归纳证明: $f_{i,j} = \max(f_{i-1,j}, f_{i-1,j-1} + B_i - jk_i)$, 我们考虑 f 的差分数组, 若 $f_{i-1,j} - f_{i-1,j-1} < B_i - jk_i$, 那么就替换。而随着 j 增大, $f_{i-1,j} - f_{i-1,j-1}$ 减小, 而 $B_i - jk_i$ 也减小, 我们要对比的是减小速度的差距! $(f_{i,j+1} - f_{i,j}) - (f_{i,j} - f_{i,j-1})$ 是逐渐增大的, 达到某个位置之后就把减速为 k 的 $B_i - jk_i$ 吊打的, 所以替换的都是一段后缀! 归纳可以证明! 其实也就是把递推式, 展开成 \sum 差分的形式, 然后两式作差, 能得到这个差分逐渐增大的形式。

Treasure Hunt

定义序列 b_1, b_2, \dots, b_c 的 beauty 值为 $\sum_{i=1}^q b_i + \sum_{i=s}^t b_i$ 的最大值, 其中 q, s, t 均为非负整数, 且 $s > q$ 或 $t \leq q$ 。注意当 $i < 1$ 或 $i > c$ 时 $b_i = 0$; 当 $s > t$ 时 $\sum_{i=s}^t b_i = 0$ 。

例如, $b = [-1, -2, -3]$ 时, beauty 值为 $0 + 0 = 0$, 此时一组可行的 q, s, t 为 $q = 0, s = 3, t = 2$ 。

$b = [-1, 2, -3]$ 时, beauty 值为 $1 + 2 = 3$, 此时一组可行的 q, s, t 为 $q = s = t = 2$ 。

给出长度为 n 的序列 a , 求出 a 的所有非空子序列 a_l, a_{l+1}, \dots, a_r ($1 \leq l \leq r \leq n$) 的 beauty 值之和。

我们发现其实就是求最大前缀和、最大子段和, 的和, 随便分治一下二维数点就好。

哥德尔机

<https://www.luogu.com.cn/blog/user3296/yi-zhong-zi-bian-cheng-ji-tong-ge-de-er-ji>

如何消除人类的必要性?

自指、不可区分性、 $O()$ -最优性。

