Task 1：我的世界

求最大值的最小值显然二分，转化为判断能否让所有的 t 都 $\le mid$。

而让一个 t_i 减到 mid 以下，需要使 t_i 被减掉 x 的次数 k 满足 $t_i - kx - s \le mid$，即 $k \ge \lceil \frac{\max(0, t_i - s - mid)}{x} \rceil$，于是要判断即为是否 $\sum\limits_{i=1}^{n} \lceil \frac{\max(0, t_i - s - mid)}{x} \rceil \le s$ 事先把 t_i 从大到小排序后可以二分找到最后一个满足 $t_m - s - mid \le 0$ 的 m，这时有：

$$\sum_{i=1}^{n} \lceil \frac{\max(0, t_i - s - mid)}{x} \rceil = \sum_{i=1}^{m} \lceil \frac{t_i - (s + mid)}{x} \rceil = \sum_{i=1}^{m} (\lceil \frac{t_i}{x} \rceil - \lceil \frac{s + mid}{x} \rceil + [t_i \bmod x > (s + mid) \bmod x])$$

上式前两项是容易求的，第三项即为求有多少个 i 满足 $i \le m$ 且 $t_i \bmod x > (s + mid) \bmod x$，主席树维护即可，$\mathcal{O}(n \log n + m \log t \log n)$。

Task 2：仰望星空

为方便，转化成选出若干个点，使得没有任何不含障碍物的子矩形包含多余一个点，并且选出的点价值之和最大。

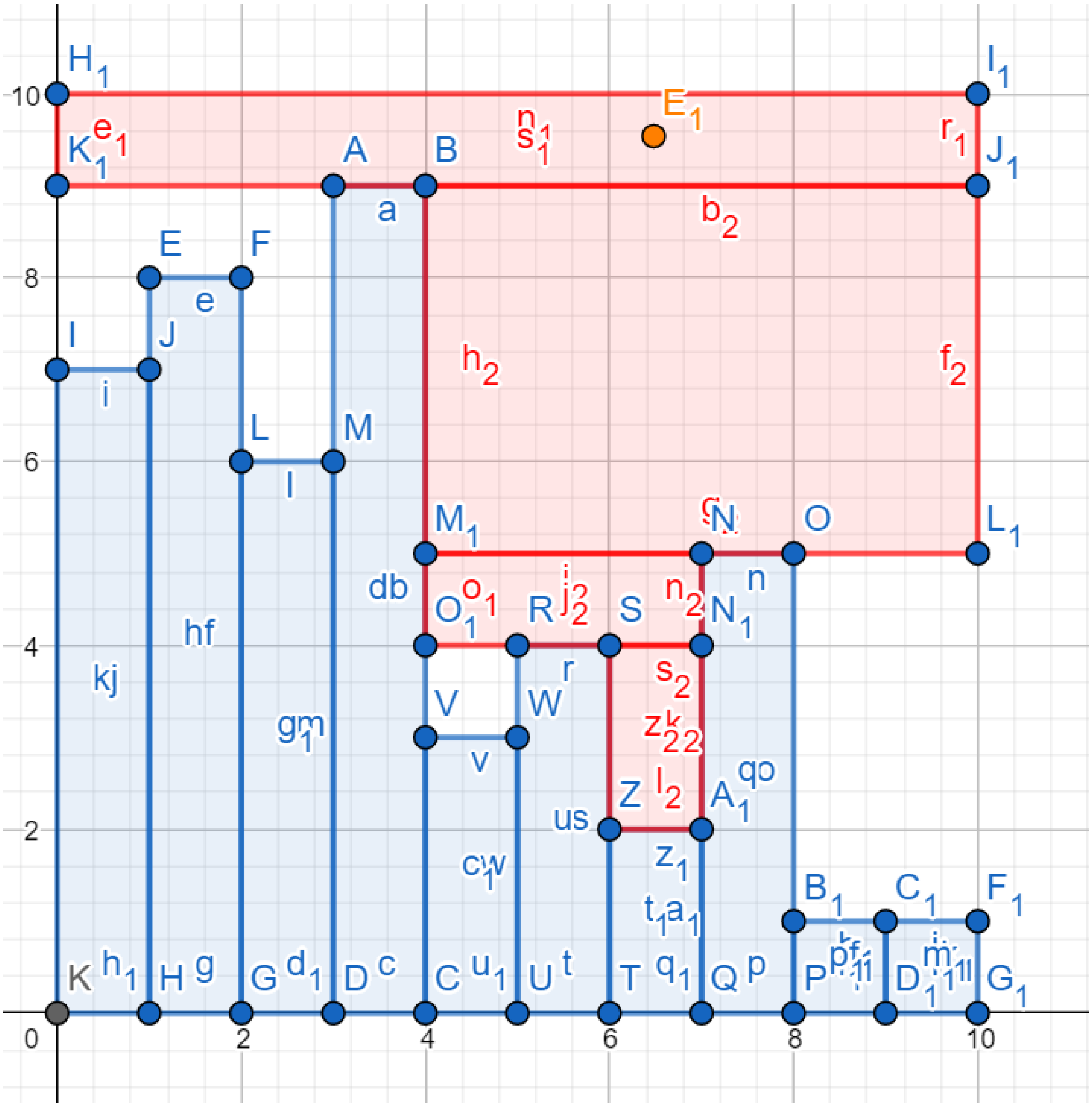
先建出 A 的笛卡尔树，最大值 A_x 对应的 x 为根，子区间 $[1, x-1]$ 和 $[x+1, n]$ 递归构建出的笛卡尔树分别为 x 的左右儿子。

设 h_x 为 x 父亲节点的 A 值（如果 x 为根则 $h_x = N$），l_x 和 r_x 分别表示 x 子树内编号最小和最大点（换句话说，x 的子树表示 A 序列的区间 $[l_x, r_x]$）定义 f_x 为只考虑子矩形 $[l_x, r_x] \times [1, h_x]$，选出 $[l_x, r_x] \times [1, h_x]$ 内的一些点使得 $[l_x, r_x] \times [1, h_x]$ 没有任何子矩形包含多于一个点，选出的点的最大价值之和。

显然 $[l_x, r_x] \times [A_x + 1, h_x]$ 是一个空矩形（下面定义为 x 的上方矩形），其中最多只能保留一个点。

如果该矩形不保留任何点，那么剩下的部分被 $x \times [1, A_x]$ 这一条障碍物完全隔开成左右两边，所以两边是独立的，把 f_{lson} 和 f_{rson} 的结果加起来即可。

如果该矩形中保留的点位于第 i 列，那么这个保留的点会限制其他位置的上方矩形里不能放任何点。具体地，如果 y 位于笛卡尔树 x 到 i 的路径上，那么 y 的上方矩形 $[l_y, r_y] \times [A_y + 1, h_y]$ 内是不能放任何点的。如下图，蓝色部分为障碍，保留点 E_1 之后，红色的子矩形内不能再放其他点：



容易从上图直观地看出，剩下有白色矩形的部分 $[1,3]\times[1,9]$、$[5,5]\times[1,4]$、$[9,10]\times[1,5]$ 彼此独立，可以把它们的 `DP` 值加起来，再加上 x 的上方矩形内保留的点 E_1 的价值即可更新 f_x。

现在要求出对于笛卡尔树 x 到 i 的路径上的所有点 y，求出 y **不在这条路径上的儿子的** `DP` 值之和。

设 f'_u 为 u 兄弟节点的 f 值（如果没有则为 0），要求的就是 x 到 i 的路径（不含 x，含 i）上所有点的 f' 值加上 i 的左右子树的 f 值。这里需要实现单点加值和路径求和，可以使用树剖，或者把单点加值路径求和转化成子树加值单点查询后用 `BIT` 来实现，$\mathcal{O}(n\log n)$。

Task 3：树的计数 II

先把置换分解成若干个轮换。

I. 连在轮换内部的边

下面给出几个结论：

`i. 1` 设一个长度为 x 的轮换上的点依次为 $a_0,a_1,\cdots,a_{x-1}$，则当 $y\in[1,x)$ 且 $2y\neq x$ 时不可能有边 (a_0,a_y)。

> 如果有边 (a_0,a_y)，那么必然对于所有的 $0\le i\le x-1$ 有边 $(a_i,a_{(i+y)\bmod x})$，由 $2y\neq x$ 得对任意 $0\le i,j\le x-1$，都有 $(i,(i+y)\bmod x)\neq((j+y)\bmod x,j)$，于是这 x 个点之间连了 x 条不同的边，一定会出现环，所以边 (a_0,a_y) 一定不存在。

> 实际上这也说明了长度为奇数的轮换一定没有内部边，长度为偶数的轮换一定没有内部边或者有且只有形如 $(a_0,a_{\frac{x}{2}})(a_1,a_{\frac{x}{2}+1})\cdots(a_{\frac{x}{2}-1},a_x-1)$ 的内部边。

`i. 2` 只有长度为 2 的轮换可能有内部边。

> 如果长度为 $2x$ （$x>1$）的轮换 $a_0,a_1,\cdots,a_{2x-1}$ 有内部边，那么由结论 `i. 1` 得该轮换的内部边为 $(a_0,a_x)(a_1,a_{x+1}),\cdots,(a_{x-1},a_{2x-1})$，然而这些边是不足以连通这 $2x$ 个点的，对于其中两个点 a_i 和 a_j，只要 $(i+x)\bmod 2x\neq j$，a_i 和 a_j 就不足以被这些边连通，不失一般性可以设 a_i 和 a_j 之间存在一条不经过这个轮换中其他点的路径 $a_i,u_1,u_2,\cdots,u_m,a_j$，这样一来 $a_{(i+x)\bmod 2x},p^x(u_1),p^x(u_2),\cdots,p^x(u_m),a_{(j+x)\bmod 2x}$ 也是 $a_{(i+x)\bmod 2x}$ 到 $a_{(j+x)\bmod 2x}$ 的一条不经过该轮换中其他点的路径，再加上 $(a_i,a_{(i+x)\bmod 2x})$ 和 $(a_j,a_{(j+x)\bmod 2x})$ 这两条边，就会形成环。

`i. 3` 最多有一个轮换有内部边。

> 如果有两个轮换 (x_0,x_1) 和 (y_0,y_1) 都存在内部边，不失一般性可以设 x_0 到 y_0 存在一条不经过 x_1 和 y_1 的路径 $x_0,u_1,u_2,\cdots,u_m,y_0$，那么 x_1 到 y_1 一定也存在一条不经过 x_0 和 y_0 的路径 $x_1,p_{u_1},p_{u_2},\cdots,p_{u_m},y_1$，再加上 (x_0,x_1) 和 (y_0,y_1) 这两条边，就会形成环。

`i. 4` 如果长度为 1 的轮换存在，那么所有的轮换都没有内部边。

> 假设有一个轮换 (x_0,x_1) 存在内部边，y 一个点单独组成一个轮换，不失一般性可以设 y 到 x_0 存在一条不经过 x_1 的路径，那么 y 到 x_1 也一定存在一条不经过 x_0 的路径，再加上 (x_0,x_1) 这条边，就会形成环。

后面我们还会证明，如果长度为 1 的轮换不存在，那么一定存在一个长度为 2 的轮换有内部边。

II. 连在轮换之间的边

考虑把所有的轮换缩成一个点，新图上两个点有边当且仅当原树上对应的两个轮换之间至少有一条边。

`ii. 1` 新图一定是一棵树。

> 新图的连通性是显然的，而如果新图有轮换数 ≥ 3 的环，设这些轮换为 $m_0,m_1,\cdots,m_{k-1}$，那么任意属于轮换 m_i 的点一定有至少一条边通向属于轮换 $m_{(i+1)\bmod k}$ 的点，不妨从 m_0 中任选一点开始不断沿着通向下一个轮换中的边一直走，这样下去必然走到重复点，从而原图成环。

`ii. 2` 假设有两个轮换的长度 x,y 满足 $x<y$，且 $x\nmid y$，则这两个轮换之间一定不能连边。

> 设轮换上的点分别为 $a_0,a_1,\cdots,a_{x-1}$ 和 $b_0,b_1,\cdots,b_{y-1}$ 由题目条件得这时有边 (a_0,b_0) 必然有边 $(a_0,b_x)(a_{y\bmod x},b_0)(a_{y\bmod x},b_x)$，而 y 不是 x 的倍数说明 $y\bmod x\neq 0$，从而 $a_0,a_{y\bmod x},b_0,b_x$ 四点成环，这说明这两个轮换之间不能连边。

`ii. 3` 如果 $x\mid y$，那么在这两个轮换都没有内部边（或前者有内部边）的情况下，这两个轮换之间可以连边，且不同的连边方案数为 x。

> 假设我们连上了 (a_0,b_l)，那么相应地对于所有的 $0\leq i\leq y-1$，会连上边 $(a_i\bmod x,b_{(l+i)\bmod y})$，从而 a 中的每个点都向 b 中的 $\frac{y}{x}$ 个点连了边，且 a 中每个点连向 y 中的点集两两不交，同时因为 b_l 可以选择 b_0 到 b_{x-1} 中的任意一点（a_0 连上 b_i 和连上 $b_{i\bmod x}$ 是等价的），所以这时两轮换之间连边有 x 种方案。可以证明当 a,b 都没有内部边或者 a 有内部边的情况下，该连法不会成环。

III. 整棵树的结构（当长度为 1 的轮换存在时）

由结论 `i. 4`，这时原树只包含轮换之间的边；由结论 `ii. 1`，轮换缩点之后得到的新图还是一棵树。

设新图上每个点的点权为对应轮换的长度，那么问题转化成对于两个长度分别为 x, y 的轮换（$x \le y$），如果 $x \mid y$，则可以在新图上对应的两点之间连边，连这条边的方案数为 x（可以看成给这条边染 $0, 1, \cdots, x-1$ 中的任意一种颜色），并且在新图上连这条边相当于原图对应的轮换之间连了 y 条边。

`iii. 1` 若长度为 1 的轮换存在，则原图连成一棵树当且仅当新图以一个点权为 1 的点为根时，新图每个点的点权都是其父亲点权的倍数。

> 若长度为 1 的轮换存在，则在新图上连成一棵树的情况下，原图一定是连通的（可以一个长度为 1 的轮换作中转点），所以这时原图连成一棵树当且仅当原图有恰好 $n-1$ 条边。
>
> 以长度为 1 的轮换为根，那么每条边 (fa_a, a) 贡献原图的边数为 $\max(x_{fa_a}, x_a)$，而所有点权之和为 n，从而有 $n-1 = \sum\limits_{a \text{ is not root}} x_a \le \sum\limits_{a \text{ is not root}} \max(x_{fa_a}, x_a)$，显然原图是树当且仅当等号成立，即对每个不为根的 a 都有 $x_{fa_a} \le x_a$，即 $x_{fa_a} \mid x_a$。

结论 `iii. 1` 说明，这时原问题在新树上等价于，首先所有点权为 1 的点构成一个连通子树，然后按点权 x 从小到大考虑，先把点权为 x 的所有点建成有根森林（每条边有 x 种染色方案），然后为森林中每一个根找一个点，该点点权 y 满足 y 是 x 的真约数，把根连向这个点，所连出的边有 y 种染色方案。

IV. 整棵树的结构（当长度为 1 的轮换不存在时）

`iv. 1` 若长度为 1 的轮换不存在，则原图恰好有一个长度为 2 的轮换有内部边。

> 类似于 `iii. 1` 可证，新图如果以点权最小的点为根，则每个点的点权都是其父亲点权的倍数（否则原图的边数一定会 $\ge n$）。
>
> 这时原图在轮换之间连的总点数为 $\sum\limits_{a \text{ is not root}} x_a = n - x_{\text{root}} < n-1$，于是只有轮换之间的边不足以使原图成树，必须要选一个长度为 2 的轮换作为根。

同理也可得，这时原问题在新树上等价于，首先所有点权为 2 的点构成一个有根连通子树（根就是有内部边的轮换），并且该有根连通子树中每条边的染色方案数为 2，后面的操作和 `III` 一样，按点权 x 从小到大考虑，先把点权为 x 的所有点建成有根森林（每条边有 x 种染色方案），然后为森林中每一个根找一个点，该点点权 y 满足 y 是 x 的真约数，把根连向这个点，所连出的边有 y 种染色方案。

V. 计数

终于到了主要部分。下面问题统一在新树上考虑。

首先点权为 1 的点（`III`）和点权为 2 的点（`IV`）所构成的无根或有根连通子树的方案数是容易计算的，对于 `III`，如果点权为 1 的点数为 d，则方案数显然为 d^{d-2}；对于 `IV`，如果点权为 2 的点数为 d，则方案数为 $d^{d-1} \times 2^{d-1}$，其中 2^{d-1} 表示对这 d 个点之间的 $d-1$ 条边染色的方案数。

后面的过程 `III` 和 `IV` 是一样的，也就是从小到大枚举点权 x，先把点权为 x 的所有点建成有根森林（每条边有 x 种染色方案），然后为森林中每一个根找一个点，该点点权 y 满足 y 是 x 的真约数，把根连向这个点，所连出的边有 y 种染色方案。

设 f_x 表示点权为 x 的真约数的点的点权之和，那么对于有根森林中的一个根，在点权为 x 真约数的所有点中找父亲连边并染色的总方案数就是 f_x。

于是问题就转化成把 d（点权为 x 的点数）个点建成有根森林，根的乘积贡献为 f_x，森林中一条边的乘积贡献为 x。

枚举根的个数 k，那么森林中的边数就是 $d-k$，这种情况对答案的贡献就是把 d 个点建成 k 棵有根树的方案数乘上 $f_x^k\times x^{d-k}$。

现在要求把 d 个点建成 k 棵有根树的方案数，考虑组合意义，相当于有一个点 $d+1$ 发出 k 条边连向这 k 个根，这就等价于把 $d+1$ 个点建成一棵无根树，使得 $d+1$ 号点的度数为 k。这相当于 `Prufer` 序列中 $d+1$ 出现了恰好 $k-1$ 次，方案数为 $\binom{d-1}{k-1}d^{d-k}$。

把所有点权的方案数乘起来即得答案。总复杂度 $\mathcal{O}(n\log n)$，瓶颈在于求 f。