

A. 简单的题

先考虑不加任何限制的情况下满足条件的方案数。

L 不是 G 的倍数的情况答案一定是 0，下面只考虑 L 是 G 的倍数的情况。

将 L 唯一分解为 $p_1^{a_1} p_2^{a_2} \dots p_k^{a_k}$ ，把 G 唯一分解为 $p_1^{b_1} p_2^{b_2} \dots p_k^{b_k}$ ，那么 $\forall 1 \leq i \leq k, a_i \geq b_i$ 。称 a_i, b_i 分别为质因子 p_i 的上下界。

显然一个集合以 G 为最大公约数，以 L 为最小公倍数等价于对于每个质因子都至少有一个数碰到上界，至少有一个数碰到下界。

那么就可以拿一个长为 $2k$ 的二进制状态表示每个质因子是否已触其上界，是否已触其下界。

直接 $O(\sigma_0(L) \times 2^{2k})$ DP 即可获得 $Q = 0$ 的全部分数。（ $\sigma_0(L)$ 为 L 的约数个数）

考虑强制选一个数的情况。

不难想到用前缀和和后缀和来合并出答案，时间复杂度 $O(\sigma_0(L) \times 3^{2n})$ 。

时间复杂度瓶颈在于枚举子集，使用高维前缀和 $O(2k \times 2^{2k})$ 预处理即可，单次合并复杂度 $O(2^{2k})$ 。

时间复杂度 $O(\sigma_0(L) \times 2^{2k})$ 。

B. 还是序列

枚举右端点，显然，满足 $\min(S_i) \leq |S|$ 的左端点是一段前缀，且随右端点右移单调右移。

接下来考虑 $|S| \leq \max(S_i)$ 的限制。

考虑笛卡尔树分治，设区间 $[L, R]$ 的最大值出现在 mid ，考虑左端点和右端点分别在 mid 两侧的段。

1. 若 $mid - l \leq r - mid$ ，在 $[L, mid]$ 中枚举左端点，向 $[mid, R]$ 中的一段做贡献，用 BIT 实现。

2. 若 $mid - l > r - mid$ ，在 $[mid, R]$ 中枚举右端点，计算 $[L, mid]$ 中的一段的贡献，用前缀和实现。

时间复杂度 $O(n \log^2 n)$ 。

C. 辉辉咖啡

设 $sum_i = \sum_{j=1}^i a_j$ ，则前 n 个限制 $\sum_{j=i}^{p_i} a_j \geq b_i$ 等价于 $sum_{i-1} - sum_{p_i} \leq -b_i$ ，可以运用差分约束系统建图 $(p_i, i-1, -b_i)$ ，那么这是一棵 $n+1$ 个点的以 n 为根的外向树。

后 m 个限制 $\sum_{j=x_i}^{y_i} a_j \leq c_i$ 等价于 $sum_{y_i} - sum_{x_i-1} \leq c_i$ ，可以建边 (x_i-1, y_i, c_i) ，由于题目保证了存在一个正整数，使得 $[x_i, y_i]$ 这个区间可以恰好被分成 k 个形如 $[s_i, p_{s_i}] (i=1, 2 \dots k)$ 的区间，所以建出的边一定是连向自己的祖先。

所以产生的负环只可能是树上节点 u 到 u 的子树内的节点 v 的链加上 v 到 u 的非树边，称 u 到 v 的链为“负环链”， v 为下端点， u 为端点。

我们要做的就是花费最小的代价使每条“负环链”至少被断掉一条边。

设 $link_x$ 为以节点 x 为下端点的“负环链”的上端点的最大深度（ n 的深度为 1），若不存在则为 0。

设 w_x 为断开 (fa_x, x) 的代价， $f_{x,i}$ 为两个端点都在 x 子树内的“负环链”以及下断点在 x 子树内且上端点深度大于 i 的“负环链”都至少被断开一条边的最小代价，转移如下：

$$(1) link_x = 0; f_{x,i} = \min \sum_{y \in son_x} f_{y,i}, w_x + \sum_{y \in son_x} f_{y,deep_x+1}$$

(2) $link_x > 0$:

$$\begin{cases} f_{x,i} = \min \sum_{y \in son_x} f_{y,i}, w_x + \sum_{y \in son_x} f_{y,deep_x+1} & i \geq deep_x \\ f_{x,i} = w_x + \sum_{y \in son_x} f_{y,deep_x+1} & i < deep_x \end{cases}$$

线段树合并维护即可。

D. 疯癫兄弟

做法 1

直接 bfs 即可，时间复杂度 $O(\alpha 2^n)$ ，其中 $\alpha = 18 + 16 + 14 + 10 + 8 + 4 + 2 = 72$ 为转移的方案数

做法 2

因为区间互不相交，所以只能在 0 的位置操作

根据哥德巴赫猜想，任意一个 ≥ 6 的偶数一定能拆成两个奇质数的和，这个猜想在 10^7 内被证明是正确的

所以长度为奇质数的一段需要操作 1 次，长度为偶数的一段需要操作 2 次，长度为非奇质数的一段要操作 3 次

找出所有连续段累加答案即可，时间复杂度线性

做法 3

手玩发现答案就是 k ，注意当 $k = 2$ 是答案为 4

做法 4

手玩发现若存在 S 段连续段，则答案就是 $2S$

做法 5

每次操作是把原本的 01 的一个长为奇质数的区间全部异或 1

考虑求原序列的差分序列，那么一次操作在差分序列上的影响就是使两个相差为奇质数的位置同时异或 1，目的就是使得差分序列全 0

增加 1 的个数一定不优，那么我们需要做的就是找到一种代价最小的匹配方案使差分数组上所有 1 两两匹配，其中两个相差奇质数长度的 1 的匹配代价为 1，两个相差偶数长度的 1 的匹配代价为 2，两个相差非质奇数长度的 1 的匹配代价为 3。可以证明，最优方案一定是奇质数的匹配最多的方案

因此只要最大化奇质数的匹配即可。发现匹配的双方一定是一个奇数和一个偶数，可以根据奇偶性建立二分图，使用 Hungary 算法即可，时间复杂度 $O(k^3)$

做法 6

使用 Dinic 求最大匹配，时间复杂度 $O(k^{2.5})$