模拟赛题解

Naganohara Yoimiya

2024.7

星天花雨（rain）

给定两个长度分别为 n, m 的 01 序列 a, b。
定义 $c_{i,j} = a_i \times b_j$ $(1 \le i \le n, 1 \le j \le m)$，问矩阵 c 中有多少个子矩形满足其中 1 的个数恰好为 k。
$1 \le n, m \le 10^5, 1 \le k \le 10^9$。

Solution

考虑 c 中以 (x_1, y_1) 为左上角，(x_2, y_2) 为右下角的子矩形内 1 的个数，发现就是 $a[x_1 \cdots x_2]$ 中 1 的个数乘上 $b[y_1 \cdots y_2]$ 中 1 的个数。

Solution

考虑 c 中以 (x_1, y_1) 为左上角，(x_2, y_2) 为右下角的子矩形内 1 的个数，发现就是 $a[x_1 \cdots x_2]$ 中 1 的个数乘上 $b[y_1 \cdots y_2]$ 中 1 的个数。
枚举 $k = x \times y$，计算 a 中有多少个子区间的和为 x，b 中有多少个子区间的和为 y 并相乘贡献到答案中即可。

Solution

如何计算 a 中有多少个子区间的和为 x?

Solution

如何计算 a 中有多少个子区间的和为 x?
求出前缀和 S，那么 $[l,r]$ 的区间和为 x 当且仅当 $S_r - S_{l-1} = x$。
枚举 r 并维护一个桶，存下来 w_i 表示当前满足 $l < r$ 且 $S_l = i$ 的 l 的个数即可。求一次的时间复杂度是 $O(n)$。

Solution

如何计算 a 中有多少个子区间的和为 x?
求出前缀和 S，那么 $[l,r]$ 的区间和为 x 当且仅当 $S_r - S_{l-1} = x$。
枚举 r 并维护一个桶，存下来 w_i 表示当前满足 $l < r$ 且 $S_l = i$ 的 l 的个数即可。求一次的时间复杂度是 $O(n)$。
由于需要对 k 的每个约数都算一遍，总的复杂度是 $O(n \times d(k))$，其中 $d(k)$ 表示 k 的约数个数。

野火（wildfire）

题面略。

Solution

考虑 $L = 0$ 的时候怎么做，也就是对一个给定的图和边权怎么算答案。

Solution

考虑 $L = 0$ 的时候怎么做，也就是对一个给定的图和边权怎么算答案。先考虑 $m = n - 1$ 也就是树的情况，这个时候一条边都不能断，所以答案就是所有 s_i 的最小值。

Solution

考虑 $L=0$ 的时候怎么做，也就是对一个给定的图和边权怎么算答案。先考虑 $m=n-1$ 也就是树的情况，这个时候一条边都不能断，所以答案就是所有 s_i 的最小值。
当 $m=n$ 的时候图是基环树，这个时候环上可以在断一条边之后立马选另一条边立刻接回来，所以我们找到环上边权最小的和次小的边 s_i, s_j，那么答案应该是 $\min(s_i+s_j, \min_{k\neq i,k\neq j} s_k)$。

Solution

那么 $L \neq 0$ 的时候就只需要二分答案然后判定就行了。

Solution

那么 $L \neq 0$ 的时候就只需要二分答案然后判定就行了。
具体来说，设二分的答案为 mid，有三种情况：

- 对于每条树边 i，他的边权必须达到 mid，于是对代价的贡献为 $\max(mid - d_i, 0)$。
- 对于环上边 i，如果他不是最小边或者次小边，那么他也需要达到 mid，贡献为 $\max(mid - d_i, 0)$。
- 对于环上的最小边和次小边 i, j，只需要 $d_i + d_j \geq mid$，故对代价的贡献为 $\max(mid - d_i - d_j, 0)$。

只要总代价 $\leq L$ 就有解。

Solution

那么 $L \neq 0$ 的时候就只需要二分答案然后判定就行了。
具体来说，设二分的答案为 mid，有三种情况：

- 对于每条树边 i，他的边权必须达到 mid，于是对代价的贡献为 $\max(mid - d_i, 0)$。
- 对于环上边 i，如果他不是最小边或者次小边，那么他也需要达到 mid，贡献为 $\max(mid - d_i, 0)$。
- 对于环上的最小边和次小边 i, j，只需要 $d_i + d_j \geq mid$，故对代价的贡献为 $\max(mid - d_i - d_j, 0)$。

只要总代价 $\leq L$ 就有解。
做一遍 DFS 预处理出环边和树边后二分即可。复杂度 $O(n \log L)$。
注意答案最大可以达到 3×10^9，需要开 long long。

赤玉琉金（blossom）

题面略。

Solution

考虑枚举一对 $i < j$，算它们对答案的贡献。

Solution

考虑 i 右侧的第一个特殊点和 j 左侧的第一个特殊点，那么不论怎么删，i, j 在最优情况下一定是通过这两个特殊点来到达的，所以它们的贡献只与这段的节点数有关。

Solution

考虑 i 右侧的第一个特殊点和 j 左侧的第一个特殊点，那么不论怎么删，i, j 在最优情况下一定是通过这两个特殊点来到达的，所以它们的贡献只与这段的节点数有关。

设这一段有 x 个节点，那么能够让 i, j 在第 k 个时刻仍然连通的排列个数就是 $\binom{n-x}{k}k!(n-k)!$，因此它对答案的贡献是

$$
\begin{aligned}
&\sum_{k=0}^{n-x}\binom{n-x}{k}k!(n-k)! \\
=&\sum_{k=0}^{n-x}\binom{n-k}{x}(n-x)!x! \\
=&(n-x)!x!\binom{n+1}{x+1} = \frac{(n+1)!}{x+1}
\end{aligned}
$$

Solution

我们考虑计算出中间有 x 个节点的 (i,j) 有多少对，如果有 c_x 对，那么给答案加上 $c_x \times \frac{(n+1)!}{x+1}$。

Solution

我们考虑计算出中间有 x 个节点的 (i,j) 有多少对，如果有 c_x 对，那么给答案加上 $c_x \times \frac{(n+1)!}{x+1}$。
注意到特殊点将序列划分成了若干段，于是我们枚举每两段做贡献，贡献形如两段等差数列加一段区间加。通过两次差分可以 $O(1)$ 处理，最后做两次前缀和即可。

Solution

我们考虑计算出中间有 x 个节点的 (i,j) 有多少对，如果有 c_x 对，那么给答案加上 $c_x \times \frac{(n+1)!}{x+1}$。
注意到特殊点将序列划分成了若干段，于是我们枚举每两段做贡献，贡献形如两段等差数列加一段区间加。通过两次差分可以 $O(1)$ 处理，最后做两次前缀和即可。
但是枚举段复杂度还是很差，但是注意到所有段的长度和 $= n$，且每对段的贡献只与其各自的段长有关。
注意到 $1+2+\cdots+m = O(m^2) = O(n)$，故本质不同段长只有 $O(m) = O(\sqrt{n})$ 种，暴力枚举即可。

Solution

我们考虑计算出中间有 x 个节点的 (i,j) 有多少对，如果有 c_x 对，那么给答案加上 $c_x \times \frac{(n+1)!}{x+1}$。
注意到特殊点将序列划分成了若干段，于是我们枚举每两段做贡献，贡献形如两段等差数列加一段区间加。通过两次差分可以 $O(1)$ 处理，最后做两次前缀和即可。
但是枚举段复杂度还是很差，但是注意到所有段的长度和 $= n$，且每对段的贡献只与其各自的段长有关。
注意到 $1+2+\cdots+m = O(m^2) = O(n)$，故本质不同段长只有 $O(m) = O(\sqrt{n})$ 种，暴力枚举即可。
时间复杂度为 $O(n)$。

致以无瑕之人（true）

题面略。

Solution

考虑一个 $f(n)$ 怎么算，发现需要 DP，设 f_i 表示能否让两个集合的差变成 i，那么每次新加入一个数字 c 有转移 $f_i \to f_{|i-c|}, f_i \to f_{i+c}$。

Solution

考虑一个 $f(n)$ 怎么算，发现需要 DP，设 f_i 表示能否让两个集合的差变成 i，那么每次新加入一个数字 c 有转移 $f_i \to f_{|i-c|}, f_i \to f_{i+c}$。
注意到当 $k \geq 9$ 的时候一定是所有数都符合条件（考虑贪心，每次往较小的一方加入当前这个数，那么差值永远不会超过 9），因此我们只需要维护 $f_{0\cdots 90}$。

Solution

经过爆搜发现，18 位数所能到达的 f 的种类数不超过 2×10^4，那就可以很轻松的做 DP 套 DP 了。f 可以直接拿一个 int128 来维护。

Solution

经过爆搜发现，18 位数所能到达的 f 的种类数不超过 2×10^4，那就可以很轻松的做 DP 套 DP 了。f 可以直接拿一个 int128 来维护。

由于有多组询问，考虑预处理 $g(i, x, S)$ 表示还要填 i 位，限制答案不能超过 x，当前状态为 S，有多少种方案。那么单次询问（显然先差分掉）如果查询的是 A 就只需要在 DFA 上依次走一遍 A 的每一位就行。