

NOIP2021 模拟赛

(2021.10.14 8:00~12:30)

解题报告

Problem A

按边权从大到小考虑每条边, 同时维护 k 个并查集来维护每轮选出的边的连通性. 对于一条边, 其删除时间是最小的 t , 使得在第 t 个并查集里 (u, v) 不连通.

注意到如果在第 t 个并查集里 (u, v) 不连通, 那么第 $t+1$ 个并查集里这两个点也是不连通的, 证明可以使用归纳法. 对于每条边二分求出最小的 k 即可.

Problem B

首先特判全白的情况. 记 c 表示至少有一个白格的列个数. 注意到以下事实: 在某一行变成全黑之前, c 不会减少; 在这之后可以每次将 c 减少 1.

因此我们的目的是要尽量快将某一行变成全黑. 枚举一行 r , 记 x 表示这一行白格的数目.

- 如果第 r 列有一个黑格, 则可以用恰好 x 次操作将第 r 行变成全黑.
- 如果第 r 列全是白色, 则我们需要额外的一次操作在第 r 列造一个黑格, 总操作数是 $x+1$.

答案即为 c 加上所有行里最小的 x 或 $x+1$.

Problem C

对于一种乘积为 x 的方案, 将所有 a_i 变成 $\frac{n}{a_i}$, 乘积将变为 $\frac{n^{2m}}{x}$. 这启示我们乘积小于 n^m 的方案数和乘积大于 n^m 的方案数一样多.

因此只需计算乘积恰好为 n^m 的方案数即可. 每个质因数是独立的, 之后就是经典的背包问题了.

Problem D

求树链 (关键点到根的链) 的并有一个经典方法: 将所有关键点按照 DFS 序排序, 记为 v_1, v_2, \dots, v_k , 则答案为

$$\sum_{i=1}^k \text{depth}(v_i) - \sum_{i=1}^{k-1} \text{depth}(\text{LCA}(v_i, v_{i+1})).$$

在本题中, 我们需要维护一个集合, 支持插入元素, 查询两个数的 LCP (最长公共前缀).

由于数非常大, 无法直接比较, 因此用线段树维护 01 串的哈希值, 可以做到 $O(\log n)$ 时间的快速比较和查询 LCP.

在本题中需要将线段树可持久化. 因为每次最多新增一个 1, 而每个 1 至多产生一次进位, 因此可以一位一位暴力修改直到当前位是 0 (此时进位停止).