重庆市第十三届程序设计大赛题解

2025 年 5 月 10 日

重庆大学（虎溪校区）

A - 题目背景是 GPT 给的

题目大意

给定长度为 n ($1 \le n \le 10^5$) 的循环数组 A，第 i 次操作对所有数同时 $A_j \leftarrow A_j | A_{i+j \bmod n}$，问多少次让 A 的数全部相同。

A - 题目背景是 GPT 给的

考虑每一位，最坏情况是这个序列的这一位只有一个 1，其他都是 0。要求这个 1 传递到其他所有位置需要的最小时间。

第一次操作后，会出现两个 1，第二次操作后，会出现 4 个 1，第 i 次操作后，会出现 $1+1+2+3+...+i$ 个 1。

因此次数是 $O(\sqrt{n})$ 的。暴力模拟操作即可，总时间复杂度 $O(n\sqrt{n})$。

B - 列车

题目大意

给 m ($1 \leq m \leq 2 \times 10^5$) 辆列车，每辆列车给运营区间 $[L, R]$ 和载客量 c，只能从 L 上车，但可以提前下车，问从 $1 \to n$ ($1 \leq n \leq 10^9$) 最多能运多少人。

B - 列车

答案为 $\min\limits_{i=1}^{n-1}\left\{\sum\limits_{j=1}^{m}c_j\times[L_j\le i<R_j]\right\}$，即经过了站 i 但不以 i 为终点的列车数的最小值。

证明：L_i 向 R_i 连边，容量为 c_i，但由于可以提前下车，$i+1$ 向 i 连边，容量为无限大，给与反悔提前下车的机会。根据最大流最小割定理，可以证明最多的人数就是最小割。

C - 区域划分

题目大意

把 $n \times n$ 的格子图划分成 n 个区域，同区域同色，相邻区域异色，使得最小色数最大。

C - 区域划分

显然答案上界是 n。现在构造一种划分方式使最小染色数是 n。

只需要任意两个区域是相邻的即可。我们把 $(1,2),(1,3),,...,(2,3),...,(n-1,n)$ 这些 1×2 的长方形放入 $n\times n$ 的格子图中即可。

D - 黄金替罪羊

题目大意

给一个长度是 $2n$ $(1 \leq n \leq 50)$ 的只有 L R？操作序列。第一个人先操作完 $1 \sim n$，然后第二个人出现在原点。接下来第一个人操作 $n+1 \sim 2n$，第二个人操作 $1 \sim n$。把？替换为 L R 求这个过程中两人不相遇的替换方案数。

D - 黄金替罪羊

先枚举第二个人出现的时候，第一个人的位置 p。用 $f_{i,j,k}$ 表示把 $1 \sim i$ 和 $n+1 \sim n+i$ 之间的？都替换了，并且用 $1 \sim i$ 可以走到 j，用 $n+1 \sim n+i$ 可以走到 k 的方案数。枚举第 i 和第 $n+i$ 位置的？换成什么得到转移。由于两个人的位置不能重合，所以把使得 $j+p=k$ 的 $f_{i,j,k} \leftarrow 0$。

最终的答案就是 $\sum_{p} \sum_{k} f_{n,p,k}$。枚举 p 是 $O(n)$ 的，f 的状态数是 $O(n^3)$ 的，转移是 $O(1)$ 的。总复杂度 $O(n^4)$。

E - 异或问题

题目大意

给五个非负整数 n, x, y, l, k ($0 \le n \le 10^9$)，求一个 a，使得 $[a, a+l-1] \oplus x$ 与 $[0, y]$ 的交集大小是 k。

E - 异或问题

一个众所周知的结论：一段连续的区间异或一个数以后，会拆分成 $O(\log n)$ 个连续的区间。

题目要求 $[a, a+l-1] \oplus x$ 中小于等于 y 的有 k 个，相当于 $[0, y] \oplus x$ 与 $[a, a+l-1]$ 的交集大小为 k。

先让 $a = 0$，然后向右逐渐移动这个区间，区间 $[a, a+l-1]$ 移动到 $[a+1, a+l]$ 的时候，区间与 $[0, y] \oplus x$ 交集的大小只会变化 $+1, 0, -1$。

我们画一个函数，横轴是 a，纵轴是区间与 $[0, y] \oplus x$ 交集的大小。这个函数只有 $O(\log n)$ 段，我们找到这些函数段的端点，就可以计算出交集大小恰好等于 k 的位置。复杂度 $O(T \log n)$。

F - 赢麻了

题目大意

根据 A,B,C 的大小关系输出对应字符串。

题解：根据 A,B,C 的大小关系输出对应字符串。注意大小写。

G - LCA & MST

题目大意

给有根树和结点的权值数组 w，定义有根树上的生成图是完全图，且 (u, v) 边的权值是 $w_{\text{LCA(u,v)}}$。求这个图的最小生成树。同时包含 q 次操作，每次操作选一个结点 p 让它和它子树的所有结点的 $w_i \leftarrow w_i + x$。每次操作完都要回答最小生成树的大小。

容易发现，除了根，每个点连向自己祖先里 w 最小的那个点得到的生成树大小是最小的。令 $f(i)$ 表示从根走到 i 这个点的路径上最小的 w。则答案为 $\sum_{i=2}^{n} f(i)$。

现在加入修改，假设现在要把 p 的子树节点的 w 加上 x。对于其中的一个结点 u 来说，若原先这个结点的祖先中 w 最小的结点在这个结点到 p 的路径上，则新的 $f(u) = \min\{f(p), f(u) + x\}$；若原先的这个结点的祖先中 w 最小的结点在 p 到根的路径上，则新的 $f(u) = f(p)$ 且此时一定有 $f(p)$ 要比 u 到 p 之间的 w 还要小，因此最终 $f(u) = \min\{f(p), f(u) + x\}$。这个过程可以用 `Segment Tree Beats` 完成。

H - 连接

题目大意

二维盒子内上边界排列有编号 $1 \sim n$ 的球，下边界也排列有 $1 \sim n$ 的球。下边界的球都紧贴下边界，上边界部分球紧贴上边界。现在要用 n 条可以弯曲的绳子把对应编号的球连接在一起，但绳子不能绕过紧贴边界的球，也不能交叉。问能不能。

H - 连接

可以连接的充要条件是紧贴上下边界的球的顺序要相同。

必要性：如果顺序不同，那么就存在上面紧贴一个 x, y，下面紧贴一个 y, x，显然无法连接。

充分性：我们不妨假设下面的球是 $1, 2, ..., n$。那么上边界紧贴的球的编号是递增的。我们先把上边界的球按 $1, 2, ..., n$ 排列，然后对应的球直接相连。由于绳子可以弯曲，只需要拖动上边界不是紧贴的球到它应该在的位置即可。

题目大意

给 n $(1 \leq n \leq 1000)$ 个点的空的 m 分图加有向边，使得所有点入度出度大于等于 k，并且任何 (u, v) 之间只能有一条边。

任意两个不同组的人都提一次问是最好的。因此先连成一个完全无向 m 分图，然后给每个边定向（提问的人指向回答的人）即可。

存在一个方案的充要条件是每个点的度数都 $\geq 2k$。必要性显然。

进行如下构造得到答案：

1. 根据握手引理，图中奇数个数的点有偶数个。假设这些点是 $1, 2, ..., 2x$，在图中增加边 $(1, 2)$,$(3, 4)$,$(5, 6)$,...,$(2x - 1, 2x)$。现在图中所有的点度数都大于等于 $2k$，并且都是偶数。

2. 在图中不断寻找回路，按照回路的方向给边赋方向，由于回路能保证每个点入一次，出一次，因此每个点入度和出度都大于等于 k。

3. 删除步骤 1 中增加的边。如果增加了边说明度数变成了至少 $2k + 2$，因此跟这个边相关的点入度和出度都大于 $k + 1$，删除一个边之后，仍然能保证入度和出度 $\geq k$。

J - RGB 树

题目大意

给 n ($1 \leq n \leq 2 \times 10^5$) 个结点的树，每个结点都被涂成了 RGB 三种颜色的一种。你要把某些结点涂白，使得：

- 任意两个非红色结点的简单路径上没有红色结点。
- 任意两个非蓝色结点的简单路径上没有蓝色结点。
- 任意两个非绿色结点的简单路径上没有绿色结点。

J - RGB 树

以蓝色为例：把蓝色的点看成障碍，可以通过涂成白色消除障碍，我们想要非蓝色的点都连通在一起。首先作为叶子的蓝色节点并不会成为障碍，我们可以不涂它，把它从树上删除，这时如果有新的蓝色节点成为叶子，就继续删除，重复这个操作，直到没有蓝色叶子，那么剩下的蓝色节点一定都要涂白。

对于另外两种颜色的点，同理。

我们发现三种颜色是互相独立的，因为考虑 x 色时，只会涂 x 色的节点。因此对原图的基础上独立进行三种颜色的计算然后答案相加即可。

K - 消消乐

题目大意

一个全是 1 的 $n \times m$ ($1 \leq n, m \leq 10^5$) 的矩阵。给行权值数组 a 和列权值数组 b ($1 \leq a_i, b_i \leq 5 \times 10^4$)。总共会进行 $n+m$ 次操作，每次操作按照剩余行列的权值为概率随机选择一行/列，把这一行/列全变为 0，并且删掉这一行/列。每次操作的代价是操作完后剩余 1 的数量，问最终的期望代价。

K - 消消乐

考虑第 i 行第 j 列的 1 对期望的贡献，即这个 1 的存活时间。我们用 R_i 表示第 i 行，第 C_i 表示第 i 列。那么这个 1 的存活时间就是在集合 $\{R_1, R_2, ..., R_n, C_1, C_2, ..., C_m\}$ 中不断按权值为概率选数，选到 R_i, C_j 中的一个就结束的期望时间。

这个期望时间就等于选数序列中出现在 R_i, C_j 前面的数的个数的期望即 $\sum_{X \neq R_i, X \neq C_j} \Pr[X \text{ selected earlier than } R_i, C_j]$。假设权值函数是 V，$V(R_i) = a_i, V(C_i) = b_i$，那么 $\Pr[X \text{ selected earlier than } R_i, C_j] = \frac{V(X)}{V(X)+V(R_i)+V(C_j)}$。

最终答案是

$$\sum_{R_i} \sum_{C_j} \sum_{\substack{X \in \{R_1,..,R_n,C_1,..,C_m\} \\ X \neq R_i, X \neq C_i}} \frac{V(X)}{V(X) + V(R_i) + V(C_j)}$$

K - 消消乐

记 $\mathrm{SCNT}(v) = \#\{(i,j) \mid V(R_i) + V(C_j) = v\}$ (即选一个列和一个行使得权值等于 v 的方案数)，$\mathrm{RCNT}(v) = \#\{i \mid V(R_i) = v\}$（即行权值为 v 的个数），$\mathrm{CCNT} = \#\{i \mid V(C_i) = v\}$（即列权值为 v 的个数）。先计算不考虑 $X = R_i, Cj$ 的情况，即先计算 $\sum_{R_i}\sum_{C_j}\sum_{X\in\{R_1,..,R_n,C_1,..,C_m\}} \frac{V(X)}{V(X)+V(R_i)+V(C_j)}$ 的答案为

$$\sum_{v\geq 0} v \times (\mathrm{CCNT}(v) + \mathrm{RCNT}(v)) \times \sum_{u\geq 0} \frac{\mathrm{SCNT}(u)}{v+u}$$

K - 消消乐

再减去 $X = R_i$ 或者 $X = C_j$ 的答案：

$$\sum_{v\geq 0}\sum_{u\geq 0}(\mathrm{CCNT}(v)\mathrm{RCNT}(u))\left(\frac{v}{v+v+u}+\frac{u}{u+u+v}\right)$$

其中 SCNT 是可以通过 a, b 中数字的出现次数的卷积得到。形如 $g(v)=\sum_{u\geq 0}\frac{f(u)}{v+u}$ 的式子求 $g(1), g(2), ..., g(n)$ 可以通过**减法卷积**计算得到。

L - 栈与重复

题目大意

对初始空栈进行 n ($1 \leq n \leq 2 \times 10^5$) 次操作：

- Push x：把 x 压入栈中。
- Pop：把栈顶弹出，不会对空栈执行这个操作。
- Repeat：重复之前的所有操作。

由于 Pop 并不会对空栈操作，所以 Repeat 操作也不会出现对空栈 Pop 的操作，因此栈会变成原来的栈再复制一份叠在上面。只需要模拟这个过程即可。

但如果一直 Repeat 会导致栈特别大，当栈大小超过 n 时，除了栈顶的 n 个元素，其他元素都没有用了，因为永远不可能 Pop 到他们，这时就不需要再复制一份叠在上面了。