

杂题选讲

whx

October 3, 2017

Description

你有一个 $n*m$ 的 int 数组 A，每次能把 $a[i]*b[j]$ 产生的一个 $n*m$ 的数组加到那个数组上去

然后最小的步数到指定的状态

$a[i]$ 和 $b[j]$ 是实数

$n, m \leq 200, A[i][j] \leq 10^6$

Solution

- 考虑把矩阵看成 n 个行向量

Solution

- 考虑把矩阵看成 n 个行向量
- 等于让你用 k 个行向量线性组合出这 n 个行向量，最小化 k

Solution

- 考虑把矩阵看成 n 个行向量
- 等于让你用 k 个行向量线性组合出这 n 个行向量，最小化 k
- 高斯消元求最大线性无关组即可，也就是矩阵的秩

Solution

- 考虑把矩阵看成 n 个行向量
- 等于让你用 k 个行向量线性组合出这 n 个行向量，最小化 k
- 高斯消元求最大线性无关组即可，也就是矩阵的秩
- 实数的精度是不够？辗转相减？求出两行的公倍数？可能爆
LL

Solution

- 考虑把矩阵看成 n 个行向量
- 等于让你用 k 个行向量线性组合出这 n 个行向量，最小化 k
- 高斯消元求最大线性无关组即可，也就是矩阵的秩
- 实数的精度是不够？辗转相减？求出两行的公倍数？可能爆
LL
- 如何少一个 $\log? \mod 10^9 + 7$ 消元，因为权值范围有限，很
大概率正确

Description

有一个 $n*m$ 的矩阵，初始每个格子的权值都为 0，可以对矩阵执行两种操作：

1. 选择一行，该行每个格子的权值加 1 或减 1。
2. 选择一列，该列每个格子的权值加 1 或减 1。

现在有 K 个限制，每个限制为一个三元组 (x,y,c) ，代表格子 (x,y) 权值等于 c 。问是否存在一个操作序列，使得操作完后的矩阵满足所有的限制。如果存在输出"Yes"，否则输出"No"。

$n, m, k \leq 1000$

Solution

- 每个限制形如 $a[i] + b[j] = c$

Solution

- 每个限制形如 $a[i] + b[j] = c$
- 注意这个限制可以由 a 唯一确定 b , 由 b 唯一确定 a 的, 也就说一个联通块内一个确定了, 别的就也确定了

Solution

- 每个限制形如 $a[i] + b[j] = c$
- 注意这个限制可以由 a 唯一确定 b , 由 b 唯一确定 a 的, 也就说一个联通块内一个确定了, 别的就也确定了
- 关系并查集维护这种『可以相互确定的关系』的联通块

Solution

- 每个限制形如 $a[i] + b[j] = c$
- 注意这个限制可以由 a 唯一确定 b , 由 b 唯一确定 a 的, 也就说一个联通块内一个确定了, 别的就也确定了
- 关系并查集维护这种『可以相互确定的关系』的联通块
- 如果两端在一个联通块内, 要么是和已有的关系等价, 要么与已有关系矛盾

Solution

- 每个限制形如 $a[i] + b[j] = c$
- 注意这个限制可以由 a 唯一确定 b , 由 b 唯一确定 a 的, 也就说一个联通块内一个确定了, 别的就也确定了
- 关系并查集维护这种『可以相互确定的关系』的联通块
- 如果两端在一个联通块内, 要么是和已有的关系等价, 要么与已有关系矛盾
- 因为是一个二分图, 所以不可能有不矛盾不等价(确定了一个联通块的值)的情况出现

Description

EP 图：点是无穷多个，边是 $(u, 2u)$ $(u, 2u + 1)$

找一条最小的包含 a 和 b 的极大反链

$$a, b \leq 10^9$$

Solution

- 考虑这个 $x \rightarrow 2x, x \rightarrow 2x+1$ 的连边方式

Solution

- 考虑这个 $x \rightarrow 2x, x \rightarrow 2x+1$ 的连边方式
- 可以试试画出来

Solution

- 考虑这个 $x \rightarrow 2x$, $x \rightarrow 2x+1$ 的连边方式
- 可以试试画出来
- 每个点入度为 1 并且没有环，所以这是一个树

Solution

- 考虑这个 $x \rightarrow 2x$, $x \rightarrow 2x+1$ 的连边方式
- 可以试试画出来
- 每个点入度为 1 并且没有环，所以这是一个树
- 这就是线段树拆节点的方式

Solution

- 线段树上找出 x 和 y 的 lca
- root 到 lca 的路径上的另一个子树都应该选上

Solution

- 线段树上找出 x 和 y 的 lca
- root 到 lca 的路径上的另一个子树都应该选上
- lca 到 x 和 y 的路径类似

Solution

■ 实现?

Solution

- 实现?
- 求出线段树上一个点的深度：每次跳到父亲， $O(\log n)$

Solution

- 实现?
- 求出线段树上一个点的深度：每次跳到父亲， $O(\log n)$
- 判定一个点能不能到达另外一个：每次跳到父亲， $O(\log n)$

Description

一个有 n 个点的无向图

求所有这样的点：

从它开始走，只要当前点有一条连出去的未访问过的边就要继续走，每次必须走一条之前没有访问过的边

如果从它开始走无论怎么走都会走出一条欧拉回路

就称它不可避

求所有不可避的点

$n \leq 10^5$

Solution

- 猜猜什么时候不可避?

Solution

- 猜猜什么时候不可避?
- 当且仅当所有环都经过这个点

Solution

- 猜猜什么时候不可避?
- 当且仅当所有环都经过这个点
- 充分性显然，必要性考虑把不经过这个点的环删去之后求欧拉回路

Solution

- 猜猜什么时候不可避?
- 当且仅当所有环都经过这个点
- 充分性显然，必要性考虑把不经过这个点的环删去之后求欧拉回路
- 怎么判定呢?

Solution

- 继续深挖性质，考虑删除这个点，所有环都过这个点的话留下的一定是森林

Solution

- 继续深挖性质，考虑删除这个点，所有环都过这个点的话留下的一定是森林
- 每次删去一个点问连通性是不是森林？

Solution

- 继续深挖性质，考虑删除这个点，所有环都过这个点的话留下的一定是森林
- 每次删去一个点问连通性是不是森林？
- 经典问题，分治维护即可

Solution

- 继续深挖性质，考虑删除这个点，所有环都过这个点的话留下的一定是森林
- 每次删去一个点问连通性是不是森林？
- 经典问题，分治维护即可
- 更优的做法？

Solution

- 考虑删去这个点之后可能分成若干联通块

Solution

- 考虑删去这个点之后可能分成若干联通块
- 每个联通块是树的充要条件是恰有 $n-1$ 条边

Solution

- 考虑删去这个点之后可能分成若干联通块
- 每个联通块是树的充要条件是恰有 $n-1$ 条边
- 如果这个点不是割点，直接减去这个点的度数

Solution

- 考虑删去这个点之后可能分成若干联通块
- 每个联通块是树的充要条件是恰有 $n-1$ 条边
- 如果这个点不是割点，直接减去这个点的度数
- 如果这个点是割点，枚举这个点连出的每条边并判断？

Solution

- 考虑删去这个点之后可能分成若干联通块
- 每个联通块是树的充要条件是恰有 $n-1$ 条边
- 如果这个点不是割点，直接减去这个点的度数
- 如果这个点是割点，枚举这个点连出的每条边并判断？
- 怎么知道连到的每个联通块所包含的边数呢

Solution

- 考虑删去这个点之后可能分成若干联通块
- 每个联通块是树的充要条件是恰有 $n-1$ 条边
- 如果这个点不是割点，直接减去这个点的度数
- 如果这个点是割点，枚举这个点连出的每条边并判断？
- 怎么知道连到的每个联通块所包含的边数呢
- 缩点双形成点双树，点双树中记录下子树边数和即可

Description

你有一段程序，共 m 行，第 i 行是 $\text{sort}(a + l[i], a + r[i] + 1)$
数组 a 的长度为 n

然后又 Q 个询问，每个询问给你一个初始的数组 $a[i]$ ，问你这个
程序能不能把 a 排序为升序

$$n \leq 1500, m \leq 10^6, q \leq 1500$$

Solution

- 先离散成一个排列

Solution

- 先离散成一个排列
- 排序问题如果我们把关注点放在每个元素能否排好序，我们就会发现，它的位置只和所有元素和它的大小关系有关

Solution

- 先离散成一个排列
- 排序问题如果我们把关注点放在每个元素能否排好序，我们就会发现，它的位置只和所有元素和它的大小关系有关
- 把 $>=d$ 的元素标记为 1，别的标记为 0，我们就得到了一个 01 串

Solution

- 先离散成一个排列
- 排序问题如果我们把关注点放在每个元素能否排好序，我们就会发现，它的位置只和所有元素和它的大小关系有关
- 把 $>=d$ 的元素标记为 1，别的标记为 0，我们就得到了一个 01 串
- 所有对应出来的 01 串排好了序那么元素就排好了序

Solution

- 先离散成一个排列
- 排序问题如果我们把关注点放在每个元素能否排好序，我们就会发现，它的位置只和所有元素和它的大小关系有关
- 把 $>=d$ 的元素标记为 1，别的标记为 0，我们就得到了一个 01 串
- 所有对应出来的 01 串排好了序那么元素就排好了序
- 01 串能否被排好序的问题可以逆序处理，把 1 都放在左边，0 都放在右边，就能获得能排好序的最差的 01 串，只要一个 01 串每个前缀的 1 的个数都小于最差情况，就一定可以被排序

Solution

- 从 01 串回到原序列，我们直接在原序列上逆序模拟即可

Solution

- 从 01 串回到原序列，我们直接在原序列上逆序模拟即可
- 原序列模拟，把每个相邻逆序对扔 set 里，就可以 $O(n^2 \log n)$ 了

Solution

- 从 01 串回到原序列，我们直接在原序列上逆序模拟即可
- 原序列模拟，把每个相邻逆序对扔 set 里，就可以 $O(n^2 \log n)$ 了
- 然后就是 check 每一个 d 的问题了，我们从小到大枚举 d，然后单点把询问的 -1，原串的 +1，维护前缀最小值

Description

你有一段程序，共 m 行，第 i 行是 $\text{sort}(a + l[i], a + r[i] + 1)$
数组 a 的长度为 n ，是一个排列
求排序后位置 i 的元素

Solution

- 还是对应到 01 串

Solution

- 还是对应到 01 串
- 对于一个 01 串，其实可以线段树模拟排序

Solution

- 还是对应到 01 串
- 对于一个 01 串，其实可以线段树模拟排序
- 找位置 i 第一次为 1 的 01 串，二分答案 + 线段树模拟排序即可

Description

有 n 个数，支持两个操作：

1. 询问 $a[l..r]$ 有多少子区间没有两个数相同
2. 修改 $a[x]$ 的值为 y

$$n \leq 10^5$$

Solution

- 考虑每个右端点最多能拓展到哪

Solution

- 考虑每个右端点最多能拓展到哪
- $f_i = \max\{pre_i | i \leq r\}$, pre_i 表示 a_i 上次出现的位置

Solution

- 考虑每个右端点最多能拓展到哪
- $f_i = \max\{pre_i | i \leq r\}$, pre_i 表示 a_i 上次出现的位置
- $ans = \sum_{i=l}^r (\max(f_i, l) - r)$

Solution

- 考虑维护 pre_i 的单调递增序列（其实就是 f_i ）

Solution

- 考虑维护 pre_i 的单调递增序列（其实就是 f_i ）
- 楼房重建

Solution

- 考虑维护 pre_i 的单调递增序列（其实就是 f_i ）
- 楼房重建
- 每个点维护这个点的单调递增序列，合并两个节点的时候右边节点取一个后缀，递归下去找出这个后缀即可

Solution

- 考虑维护 pre_i 的单调递增序列（其实就是 f_i ）
- 楼房重建
- 每个点维护这个点的单调递增序列，合并两个节点的时候右边节点取一个后缀，递归下去找出这个后缀即可
- 复杂度 $O(n \log^2 n)$

Description

$2^n - 1$ 个村庄，每个村庄 n 户人家，每户养一条狗。每个村庄都有至少一只狗得了懒癌，一共 $2^n - 1$ 种情况，假设所有情况的概率均匀分布。

一个人来到村庄中，告诉所有人：“你们村里至少有一只狗得了懒癌！”

每个人不能看出自己的狗有没有得懒癌，具体来说可以用一个 n 个结点的有向图来描述可见性， v 到 u 有一条有向边表示门牌号为 v 的人家能看出门牌号为 u 的家里的狗是否得了懒癌，没边则表示看不出。每个人都知道这张有向图。但是每个人都不会告诉别人自己观察的结果。

Description

每个人每天出门看所有他能看的狗，如果一个人推断自己的狗得了懒癌，就会在这天结束的时候一枪干掉自己的狗。

每个人都能听到枪声，如果没有听到枪声那么第二天会继续出门看狗。直到大家听到了枪声，就不会杀狗了。

求期望多少只狗会死亡，以及期望第几天会响起枪声（如果没有开枪算作 0）。 $n \leq 3000$

命题人：吉如一

Solution

- 台下有做过这道题的同学可以上来讲讲做法。

Solution

- 台下有做过这道题的同学可以上来讲讲做法。
- 不知道大家有没有听说过红眼睛蓝眼睛的故事？

Solution

- 台下有做过这道题的同学可以上来讲讲做法。
- 不知道大家有没有听说过红眼睛蓝眼睛的故事？
- 考虑第一天，每个人会开枪当且仅当他看到剩下 $n - 1$ 只狗都是正常的。

Solution

- 台下有做过这道题的同学可以上来讲讲做法。
- 不知道大家有没有听说过红眼睛蓝眼睛的故事？
- 考虑第一天，每个人会开枪当且仅当他看到剩下 $n - 1$ 只狗都是正常的。
- 第二天早上，每个人都知道没有人看到剩下 $n - 1$ 只狗都是正常的，有两种可能。

Solution

- 台下有做过这道题的同学可以上来讲讲做法。
- 不知道大家有没有听说过红眼睛蓝眼睛的故事？
- 考虑第一天，每个人会开枪当且仅当他看到剩下 $n - 1$ 只狗都是正常的。
- 第二天早上，每个人都知道自己看到剩下 $n - 1$ 只狗都是正常的，有两种可能。
- 第一，看到 $n - 1$ 只狗的人（设这个集合为 S_1 ）看到的狗有得了懒癌的。第二，没有人能看到全部的 $n - 1$ 条狗。

Solution

- 台下有做过这道题的同学可以上来讲讲做法。
- 不知道大家有没有听说过红眼睛蓝眼睛的故事？
- 考虑第一天，每个人会开枪当且仅当他看到剩下 $n - 1$ 只狗都是正常的。
- 第二天早上，每个人都看到没有人看到剩下 $n - 1$ 只狗都是正常的，有两种可能。
 - 第一，看到 $n - 1$ 只狗的人（设这个集合为 S_1 ）看到的狗有得了懒癌的。第二，没有人能看到全部的 $n - 1$ 条狗。
 - 如果是第二种，所有人没有得到额外的信息（因为所有人本来就知道这个有向图），所以永远不会开枪了。

Solution

- 如果是第一种，考虑额外信息对第二天决策的贡献。

Solution

- 如果是第一种，考虑额外信息对第二天决策的贡献。
- 第二天一个人会杀死自己的狗，一定是他看到了第一天某个人 $x \in S_1$ 看到的 $n - 1$ 条狗中，除了自己的狗之外都是健康的。

Solution

- 如果是第一种，考虑额外信息对第二天决策的贡献。
- 第二天一个人会杀死自己的狗，一定是他看到了第一天某个人 $x \in S_1$ 看到的 $n - 1$ 条狗中，除了自己的狗之外都是健康的。
- 他可能能看到 $n - 2$ 条狗，或者 $n - 1$ 条狗（如果第一天他看到的狗中，只有 x 是有懒癌的话）。

Solution

- 如果是第一种，考虑额外信息对第二天决策的贡献。
- 第二天一个人会杀死自己的狗，一定是他看到了第一天某个人 $x \in S_1$ 看到的 $n - 1$ 条狗中，除了自己的狗之外都是健康的。
- 他可能能看到 $n - 2$ 条狗，或者 $n - 1$ 条狗（如果第一天他看到的狗中，只有 x 是有懒癌的话）。
- 我们其实已经可以得出什么时候会一定会杀死自己的狗了。

Solution

- 如果是第一种，考虑额外信息对第二天决策的贡献。
- 第二天一个人会杀死自己的狗，一定是他看到了第一天某个人 $x \in S_1$ 看到的 $n - 1$ 条狗中，除了自己的狗之外都是健康的。
- 他可能能看到 $n - 2$ 条狗，或者 $n - 1$ 条狗（如果第一天他看到的狗中，只有 x 是有懒癌的话）。
- 我们其实已经可以得出什么时候会一定会杀死自己的狗了。
- 一定是存在一个不断缩小的集合链，每次一个 $x \in S_i$ 能看到 $S_i / \{x\}$ 的所有狗，然后没有开枪，说明这些狗中还有懒癌。于是集合缩小到 $S_{i+1} = S_i / \{x\}$ ，直到什么时候 S_k 里没有懒癌了。

Solution

- 如果是第一种，考虑额外信息对第二天决策的贡献。
- 第二天一个人会杀死自己的狗，一定是他看到了第一天某个人 $x \in S_1$ 看到的 $n - 1$ 条狗中，除了自己的狗之外都是健康的。
- 他可能能看到 $n - 2$ 条狗，或者 $n - 1$ 条狗（如果第一天他看到的狗中，只有 x 是有懒癌的话）。
- 我们其实已经可以得出什么时候会一定会杀死自己的狗了。
- 一定是存在一个不断缩小的集合链，每次一个 $x \in S_i$ 能看到 $S_i / \{x\}$ 的所有狗，然后没有开枪，说明这些狗中还有懒癌。于是集合缩小到 $S_{i+1} = S_i / \{x\}$ ，直到什么时候 S_k 里没有懒癌了。
- 对应到图论的语言就是，有红点和蓝点，每次删去一个能到所有点的点，直到剩下的点全部都是蓝点为止。

Solution

- 如果是第一种，考虑额外信息对第二天决策的贡献。
- 第二天一个人会杀死自己的狗，一定是他看到了第一天某个人 $x \in S_1$ 看到的 $n - 1$ 条狗中，除了自己的狗之外都是健康的。
- 他可能能看到 $n - 2$ 条狗，或者 $n - 1$ 条狗（如果第一天他看到的狗中，只有 x 是有懒癌的话）。
- 我们其实已经可以得出什么时候会一定会杀死自己的狗了。
- 一定是存在一个不断缩小的集合链，每次一个 $x \in S_i$ 能看到 $S_i / \{x\}$ 的所有狗，然后没有开枪，说明这些狗中还有懒癌。于是集合缩小到 $S_{i+1} = S_i / \{x\}$ ，直到什么时候 S_k 里没有懒癌了。
- 对应到图论的语言就是，有红点和蓝点，每次删去一个能到所有点的点，直到剩下的点全部都是蓝点为止。
- 能到所有点的点，其实就是取反有/无边情况之后，出度为 0 的点。

Solution

- 因为只需要考虑有枪声的情况，所以红点不能出现在强连通分量或是强连通分量的前驱中。

Solution

- 因为只需要考虑有枪声的情况，所以红点不能出现在强连通分量或是强连通分量的前驱中。
- 去掉这部分之后，剩下的 DAG 里，求一个拓扑序使得最后一个红点出现的尽量早。

Solution

- 因为只需要考虑有枪声的情况，所以红点不能出现在强连通分量或是强连通分量的前驱中。
- 去掉这部分之后，剩下的 DAG 里，求一个拓扑序使得最后一个红点出现的尽量早。
- 经典问题，机场调度。变成每次删一个入度为 0 的点，使得全是蓝点的前缀尽量长，蓝点能删即删即可。

Solution

- 因为只需要考虑有枪声的情况，所以红点不能出现在强连通分量或是强连通分量的前驱中。
- 去掉这部分之后，剩下的 DAG 里，求一个拓扑序使得最后一个红点出现的尽量早。
- 经典问题，机场调度。变成每次删一个入度为 0 的点，使得全是蓝点的前缀尽量长，蓝点能删即删即可。
- 于是答案等于多少个点能被至少一个红点到达，期望可加性交换求和顺序，转而考虑一个点能被至少一个红点到达的概率。

Solution

- 因为只需要考虑有枪声的情况，所以红点不能出现在强连通分量或是强连通分量的前驱中。
- 去掉这部分之后，剩下的 DAG 里，求一个拓扑序使得最后一个红点出现的尽量早。
- 经典问题，机场调度。变成每次删一个入度为 0 的点，使得全是蓝点的前缀尽量长，蓝点能删即删即可。
- 于是答案等于多少个点能被至少一个红点到达，期望可加性交换求和顺序，转而考虑一个点能被至少一个红点到达的概率。
- 变成了传递闭包问题（每个点能被多少个点到达）。

Solution

- 因为只需要考虑有枪声的情况，所以红点不能出现在强连通分量或是强连通分量的前驱中。
- 去掉这部分之后，剩下的 DAG 里，求一个拓扑序使得最后一个红点出现的尽量早。
- 经典问题，机场调度。变成每次删一个入度为 0 的点，使得全是蓝点的前缀尽量长，蓝点能删即删即可。
- 于是答案等于多少个点能被至少一个红点到达，期望可加性交换求和顺序，转而考虑一个点能被至少一个红点到达的概率。
- 变成了传递闭包问题（每个点能被多少个点到达）。
- 对于一共有多少只狗会死，其实就相当于最后只剩红点的时候顶在最前面的那些红点，也就是每个点为红点并且前面没有红点的概率之和。