贪心相关杂题选讲

mcfx

2019 年 7 月 2 日

贪心

贪心只是一种思想，小部分题目可能直接贪心后就能完成（如下面这道），大部分题目都需要贪心之后再进一步推导，或是用某些算法维护。

矩阵取数

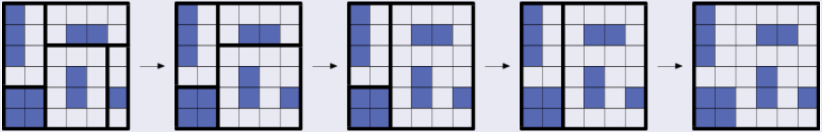
$n \times m$ 矩阵，每行取一个数，使取出的所有数总和最大。

接下来可能有一些题目有不少人见过，这种就直接跳过吧。

CF 1181 E2

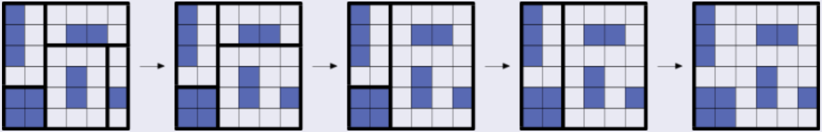
A Story of One Country (Hard)

平面上 $N(N \leq 10^5)$ 个不相交矩形，要求对每一个矩形，找到一个包含他的矩形，使得这些新的矩形可以按相邻边合并直到只剩一个矩形。



A Story of One Country (Hard)

平面上 $N(N \leq 10^5)$ 个不相交矩形，要求对每一个矩形，找到一个包含他的矩形，使得这些新的矩形可以按相邻边合并直到只剩一个矩形。

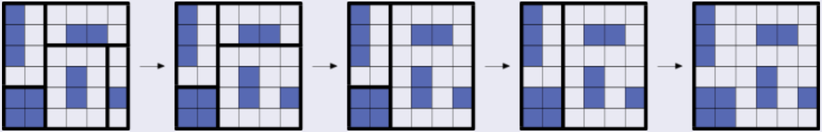


贪心的划分，如果一条线能把这些矩形划分成相离的两边，那么直接划分开肯定不劣。

CF 1181 E2

A Story of One Country (Hard)

平面上 $N(N \le 10^5)$ 个不相交矩形，要求对每一个矩形，找到一个包含他的矩形，使得这些新的矩形可以按相邻边合并直到只剩一个矩形。

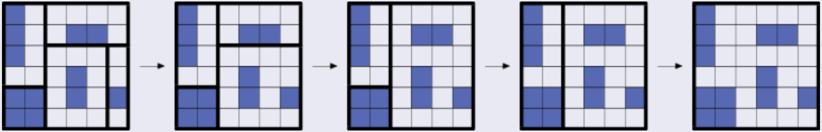


贪心的划分，如果一条线能把这些矩形划分成相离的两边，那么直接划分开肯定不劣。
对于每个方向，每个矩形可以拆成两个事件，$+1$ 和 -1，一个前缀和是 0 说明能够划分。
可以用数据结构维护，每个事件是个区间加。
每次划分时，将较小的一边暴力重建。

CF 1181 E2

A Story of One Country (Hard)

平面上 $N(N \le 10^5)$ 个不相交矩形，要求对每一个矩形，找到一个包含他的矩形，使得这些新的矩形可以按相邻边合并直到只剩一个矩形。



贪心的划分，如果一条线能把这些矩形划分成相离的两边，那么直接划分开肯定不劣。
对于每个方向，每个矩形可以拆成两个事件，$+1$ 和 -1，一个前缀和是 0 说明能够划分。
可以用数据结构维护，每个事件是个区间加。
每次划分时，将较小的一边暴力重建。
复杂度 $O(N \log^2 N)$。CF 上的题解要简单一些，可以参考。

CF 1097 E

Egor and an RPG game

一个长为 $N(N \leq 10^5)$ 的排列，要求划分成若干个不相交单调（上升或下降）子序列，个数需要不超过 K，K 是让所有长为 N 的排列都能划分的最小值。（K 没有告诉你）

CF 1097 E

Egor and an RPG game

一个长为 $N(N \leq 10^5)$ 的排列，要求划分成若干个不相交单调(上升或下降）子序列，个数需要不超过 K，K 是让所有长为 N 的排列都能划分的最小值。(K 没有告诉你)

一个简单的贪心，每次选出最长的一个排列，但是是有反例的：$[19, 10, 5, 4, 25, 28, 14, 26, 23, 17, 33, 34, 3, 20, 27, 31, 1, 7, 6, 32, 29, 12, 22, 8, 13, 2, 15, 21, 30, 24, 9, 11, 16, 18]$（较小的反例有点难找）

Egor and an RPG game

一个长为 $N(N \leq 10^5)$ 的排列，要求划分成若干个不相交单调 (上升或下降）子序列，个数需要不超过 K，K 是让所有长为 N 的排列都能划分的最小值。(K 没有告诉你)

一个简单的贪心，每次选出最长的一个排列，但是是有反例的：$[19, 10, 5, 4, 25, 28, 14, 26, 23, 17, 33, 34, 3, 20, 27, 31, 1, 7, 6, 32, 29, 12, 22, 8, 13, 2, 15, 21, 30, 24, 9, 11, 16, 18]$（较小的反例有点难找）

考虑这个排列：$[1, 3, 2, 6, 5, 4, 10, 9, 8, 7, 15, 14, 13, 12, 11]$，显然至少要用 5 个单调序列。

实际上，对于 N，K 只需要满足 $N < \frac{(K+1)\cdot(K+2)}{2}$。

每次找出 LIS，如果其长度大于 K，那么可以直接删除，剩下的数个数不超过 $N - K - 1 \leq \frac{(K+1)\cdot(K+2)}{2} - K - 1 = \frac{K\cdot(K+1)}{2}$。

否则一定能划分为 LIS 长度个下降序列。

CF 1067 D

Computer Game

有 n 个游戏，每个游戏有收益 a_i，升级后的收益 b_i，每次成功概率 p_i。每秒可以玩一个游戏，如果成功则得到当前收益，并且可以升级任意某个游戏。求 t 秒后的期望收益的最大值。

$n \le 10^5, t \le 10^{10}, a < b$

Computer Game

有 n 个游戏，每个游戏有收益 a_i，升级后的收益 b_i，每次成功概率 p_i。每秒可以玩一个游戏，如果成功则得到当前收益，并且可以升级任意某个游戏。求 t 秒后的期望收益的最大值。
$n \le 10^5, t \le 10^{10}, a < b$

设最大的 b_ip_i 为 M，显然每秒的收益不可能超过这个。那么最优策略是先玩一些其他游戏，当成功一次之后就不停玩 b_ip_i 最大的游戏。

Computer Game

有 n 个游戏，每个游戏有收益 a_i，升级后的收益 b_i，每次成功概率 p_i。每秒可以玩一个游戏，如果成功则得到当前收益，并且可以升级任意某个游戏。求 t 秒后的期望收益的最大值。
$n \le 10^5, t \le 10^{10}, a < b$

设最大的 $b_i p_i$ 为 M，显然每秒的收益不可能超过这个。那么最优策略是先玩一些其他游戏，当成功一次之后就不停玩 $b_i p_i$ 最大的游戏。
考虑 dp，dp 状态只和剩余多少时间有关，那么 dp 转移是
$dp_{t+1} = \max\{p_i(a_i + tM) + (1 - p_i)dp_t\} =$
$dp_t + \max\{p_i(tM - dp_t) + p_i a_i\}$。

CF 1067 D

Computer Game

有 n 个游戏，每个游戏有收益 a_i，升级后的收益 b_i，每次成功概率 p_i。每秒可以玩一个游戏，如果成功则得到当前收益，并且可以升级任意某个游戏。求 t 秒后的期望收益的最大值。
$n \le 10^5, t \le 10^{10}, a < b$

设最大的 b_ip_i 为 M，显然每秒的收益不可能超过这个。那么最优策略是先玩一些其他游戏，当成功一次之后就不停玩 b_ip_i 最大的游戏。
考虑 dp，dp 状态只和剩余多少时间有关，那么 dp 转移是
$dp_{t+1} = \max\{p_i(a_i + tM) + (1 - p_i)dp_t\} =$
$dp_t + \max\{p_i(tM - dp_t) + p_ia_i\}$。
可以发现形成了一个斜率的形式，那么可以建凸包之后每次二分，复杂度为 $O((n + t) \log n)$。

考虑 $dp_{t+1} - dp_t \leq M$，那么 $tM - dp_t \leq (t+1)M - dp_{t+1}$，也就是说在凸包上走的位置是单调递增的。

考虑 $dp_{t+1} - dp_t \le M$，那么 $tM - dp_t \le (t+1)M - dp_{t+1}$，也就是说在凸包上走的位置是单调递增的。
那么可以用矩阵快速幂优化 dp，每次在凸包上这一段走，倍增的找到最远能走到哪，时间复杂度 $O(n(\log n + \log t))$。

CF 1054 G

New Road Network

有 n 个点，你需要构造一棵树，满足 m 个要求，每个要求是某些点必须是个连通块。
$n, m \le 2 \cdot 10^3$

CF 1054 G

New Road Network

有 n 个点，你需要构造一棵树，满足 m 个要求，每个要求是某些点必须是个连通块。
$n, m \leq 2 \cdot 10^3$

对每个点 i，求出有哪些要求包含他，记作 A_i。

CF 1054 G

New Road Network

有 n 个点，你需要构造一棵树，满足 m 个要求，每个要求是某些点必须是个连通块。
$n, m \le 2 \cdot 10^3$

对每个点 i，求出有哪些要求包含他，记作 A_i。
实际上直接以 $|A_i \cap A_j|$ 为边权跑最大生成树，要么结果是答案，要么无解。

New Road Network

有 n 个点，你需要构造一棵树，满足 m 个要求，每个要求是某些点必须是个连通块。
$n, m \leq 2 \cdot 10^3$

对每个点 i，求出有哪些要求包含他，记作 A_i。
实际上直接以 $|A_i \cap A_j|$ 为边权跑最大生成树，要么结果是答案，要么无解。
证明可以这样考虑：

New Road Network

有 n 个点，你需要构造一棵树，满足 m 个要求，每个要求是某些点必须是个连通块。
$n, m \leq 2 \cdot 10^3$

对每个点 i，求出有哪些要求包含他，记作 A_i。
实际上直接以 $|A_i \cap A_j|$ 为边权跑最大生成树，要么结果是答案，要么无解。
证明可以这样考虑：
如果有一个要求只包含一个点，可以直接删掉。

New Road Network

有 n 个点，你需要构造一棵树，满足 m 个要求，每个要求是某些点必须是个连通块。
$n, m \leq 2 \cdot 10^3$

对每个点 i，求出有哪些要求包含他，记作 A_i。
实际上直接以 $|A_i \cap A_j|$ 为边权跑最大生成树，要么结果是答案，要么无解。
证明可以这样考虑：
如果有一个要求只包含一个点，可以直接删掉。
假设存在这样的树，那么对于一个叶子 x，如果 x 与 y 相连，那么一定 $A_x \subset A_y$。

CF 1054 G

New Road Network

有 n 个点，你需要构造一棵树，满足 m 个要求，每个要求是某些点必须是个连通块。
$n, m \le 2 \cdot 10^3$

对每个点 i，求出有哪些要求包含他，记作 A_i。
实际上直接以 $|A_i \cap A_j|$ 为边权跑最大生成树，要么结果是答案，要么无解。
证明可以这样考虑：
如果有一个要求只包含一个点，可以直接删掉。
假设存在这样的树，那么对于一个叶子 x，如果 x 与 y 相连，那么一定 $A_x \subset A_y$。
那么如果不存在 $A_x \subset A_y$ 的两个点，一定无解。否则可以每次找一个叶子删掉，直到 $n = 1$。

CF 1054 G

New Road Network

有 n 个点，你需要构造一棵树，满足 m 个要求，每个要求是某些点必须是个连通块。
$n, m \le 2 \cdot 10^3$

对每个点 i，求出有哪些要求包含他，记作 A_i。
实际上直接以 $|A_i \cap A_j|$ 为边权跑最大生成树，要么结果是答案，要么无解。
证明可以这样考虑：
如果有一个要求只包含一个点，可以直接删掉。
假设存在这样的树，那么对于一个叶子 x，如果 x 与 y 相连，那么一定 $A_x \subset A_y$。
那么如果不存在 $A_x \subset A_y$ 的两个点，一定无解。否则可以每次找一个叶子删掉，直到 $n = 1$。
实际上这个过程求出的就是一棵最大生成树。

CF 1120 F

Secret Letters

有 W 和 P 两个人，他们要互相寄 $N(N \le 10^5)$ 封信。每封信可以直接寄，花费 d 代价，或者暂存起来，暂存 T 秒需要 cT 代价。暂存的信只能当另一个人下次去暂存信时，或者最后时刻才能取。求最小代价。

CF 1120 F

Secret Letters

有 W 和 P 两个人，他们要互相寄 $N(N \leq 10^5)$ 封信。每封信可以直接寄，花费 d 代价，或者暂存起来，暂存 T 秒需要 cT 代价。暂存的信只能当另一个人下次去暂存信时，或者最后时刻才能取。求最小代价。

如果某封信是暂存的，那么之后为了取信，一定一直有信是暂存的。

倒推，对于发件人相同的一段，需要考虑是暂存还是直接发。

CF 883 J

Renovation

有 m 个建筑，每个建筑拆除需要 p_i 代价，装修需要 b_i 代价。有 n 个月，每个月预算 a_i。
每个月可以拆除若干建筑，但为了欺骗市民是装修，需要 $b_j \le a_i$ 才能拆除 j 建筑。拆除可以用到之前存的预算。
求最多拆除多少个建筑。
$n, m \le 10^5$。

Renovation

有 m 个建筑，每个建筑拆除需要 p_i 代价，装修需要 b_i 代价。有 n 个月，每个月预算 a_i。
每个月可以拆除若干建筑，但为了欺骗市民是装修，需要 $b_j \le a_i$ 才能拆除 j 建筑。拆除可以用到之前存的预算。
求最多拆除多少个建筑。
$n, m \le 10^5$。

按照 p_i 从小到大排序，每次找到能满足 b_i 的最后一个位置，并判断之前的预算够不够。
如果够的话，从这个位置往前，把若干个位置的预算减去一定值，用数据结构维护即可。

CF 865 D

Renovation

n 个时刻，每个时刻可以买入、卖出一股或者什么都不干。已知每个时刻的股价，求最大收益。
$n \leq 3 \cdot 10^5$。

Renovation

n 个时刻，每个时刻可以买入、卖出一股或者什么都不干。已知每个时刻的股价，求最大收益。
$n \leq 3 \cdot 10^5$。

对于每一天，可以在未来保留在这一天买入的机会。
也就是说，用堆维护之前买入的机会，如果当前股价低于之前的最小值，那么直接插入堆中；否则可以认为

洗衣服

你现在要洗 l 件衣服。你有 n 台洗衣机和 m 台烘干机。由于你的机器非常的小，因此你每次只能洗涤（烘干）一件衣服。
第 i 台洗衣机洗一件衣服需要 w_i 分钟，第 i 台烘干机烘干一件衣服需要 d_i 分钟。
请问把所有衣服洗干净并烘干，最少需要多少时间？假设衣服在机器间转移不需要时间，并且洗完的衣服可以过一会再烘干。

洗衣服

你现在要洗 l 件衣服。你有 n 台洗衣机和 m 台烘干机。由于你的机器非常的小，因此你每次只能洗涤（烘干）一件衣服。
第 i 台洗衣机洗一件衣服需要 w_i 分钟，第 i 台烘干机烘干一件衣服需要 d_i 分钟。
请问把所有衣服洗干净并烘干，最少需要多少时间？假设衣服在机器间转移不需要时间，并且洗完的衣服可以过一会再烘干。

把两种机器分开考虑，可以使用堆求出最小的若干个时间。把洗衣机的从小到大排序，烘干机的从大到小排序，分别相加取最大值即可。

Forgotten Tree 9

有一棵二叉树，按照中序遍历标号为 $1 \sim N$。可以询问一个点的子树对应区间是否是给定区间。要求 $3n$ 次询问之内求出树形态。

Forgotten Tree 9

有一棵二叉树，按照中序遍历标号为 $1 \sim N$。可以询问一个点的子树对应区间是否是给定区间。要求 $3n$ 次询问之内求出树形态。

维护每个已经询问出的子树。每次找出最小的没处理过的叶子。对于这样的连通块，先考虑向左扩展，再考虑向右扩展。向左扩展时如果左边只有一个点后面就是已经连通的，这部分也需要加进来一起处理。