Open Cup 趣题选讲

蒋凌宇

西南大学附属中学校

引言

Open Cup named after E.V. Pankratiev 是对世界各地举行的算法竞赛题目的收集，包括 Petrozavodsk Camp, EC-Final, Moscow Workshop 等，也有一些不太知名的高质量比赛。

Open Cup 的题目难度较大，质量普遍较高，很适合算法竞赛的后期训练。然而其门槛较高，一般只有水平较高的选手才能申请到账号 (例如入围 ICPC World Finals 的队伍)，但题面都可以在 Codeforces 上 zimpha 的博客[1]中获取。本次讲课我精选了若干道 Open Cup 中比较有趣的题目，希望能让大家开阔视野，同时对之后的训练有所帮助。

[1] https://codeforces.com/blog/entry/84466

Cactus Competition[2]

给定一张 $N \times M$ 的网格图，(i, j) 的权值为 $A_i + B_j$，其中 A_i，B_j 是给定的两数列。

求有多少对 (S, T) $(1 \le S \le T \le N)$ 满足存在一条从 $(S, 1)$ 走到 (T, M) 的路径，每一步只能往下或往右走一步，且经过的所有格子的权值非负。

$1 \le N, M \le 200\,000$
$-10^9 \le A_i, B_j \le 10^9$

1024 MB, 2 s

[2] XXI Open Cup. Grand Prix of Korea, Problem B

Cactus Competition

不妨先考虑 $S = 1, T = N$ 的情况。可以证明存在路径当且仅当下列所有条件都**不满足**：

Cactus Competition

不妨先考虑 $S = 1, T = N$ 的情况。可以证明存在路径当且仅当下列所有条件都**不满足**:

- $\min_{1 \le i \le N}\{A_i\} + \max_{1 \le j \le M}\{B_j\} < 0$。
- $\max_{1 \le i \le N}\{A_i\} + \min_{1 \le j \le M}\{B_j\} < 0$。
- 存在 $1 \le x \le N, 1 \le y \le M$ 使得 $A_x + B_j < 0\ (1 \le j \le y)$ 且 $A_i + B_y < 0\ (1 \le i \le x)$。
- 存在 $1 \le x \le N, 1 \le y \le M$ 使得 $A_x + B_j < 0\ (y \le j \le M)$ 且 $A_i + B_y < 0\ (x \le i \le N)$。

Cactus Competition

不妨先考虑 $S=1, T=N$ 的情况。可以证明存在路径当且仅当下列所有条件都**不满足**：

- $\min_{1\le i\le N}\{A_i\}+\max_{1\le j\le M}\{B_j\}<0$。
- $\max_{1\le i\le N}\{A_i\}+\min_{1\le j\le M}\{B_j\}<0$。
- 存在 $1\le x\le N, 1\le y\le M$ 使得 $A_x+B_j<0\ (1\le j\le y)$ 且 $A_i+B_y<0\ (1\le i\le x)$。
- 存在 $1\le x\le N, 1\le y\le M$ 使得 $A_x+B_j<0\ (y\le j\le M)$ 且 $A_i+B_y<0\ (x\le i\le N)$。

后两个条件会直接让一些起点和终点不可行，可以用单调栈和二分预处理。对于每个剩下的起点，前两个条件给它限定了一个区间，且区间端点是单调的，因此扫一遍即可求出答案。时间复杂度 $O(N\log N+M)$。

Query On A Tree 17[3]

给定一棵 N 个点的有根树，初始时每个点的点权为 0。

接下来会执行 Q 次操作，每次操作会是以下两种之一：

- 将 u 子树内所有顶点的点权增加 1。
- 将 u 到 v 路径上内所有顶点的点权增加 1。

在每次操作后，设顶点 u 的点权为 A_u，则输出一个顶点 v，使得 $\sum_{1\le u\le N} A_u \cdot \mathrm{dis}(u,v)$ 最小。若有多个满足条件的顶点，输出深度最小的一个。

$2 \le N \le 100\,000$
$1 \le Q \le 100\,000$

1024 MB, 2 s

[3] XXI Open Cup. Grand Prix of Korea, Problem I

Query On A Tree 17

设所有点的点权和为 $S > 0$，则最浅的带权重心的子树点权和一定大于 $S/2$。若将所有顶点按 DFS 序写下，每个点的出现次数是其点权，则重心的子树是该序列上的一个区间，且长度大于序列长度的一半，因此一定包含最中间的一个数 (若长度为偶数则可以任取一个)。

Query On A Tree 17

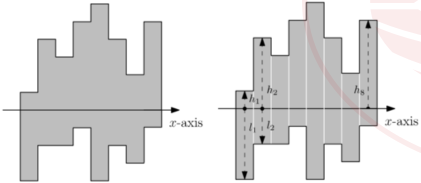
设所有点的点权和为 $S > 0$，则最浅的带权重心的子树点权和一定大于 $S/2$。若将所有顶点按 DFS 序写下，每个点的出现次数是其点权，则重心的子树是该序列上的一个区间，且长度大于序列长度的一半，因此一定包含最中间的一个数 (若长度为偶数则可以任取一个)。

用重链剖分加线段树维护，询问时在线段树上二分找到这个最中间的顶点，则最浅的重心一定在它到根的路径上，倍增查找即可。时间复杂度 $O(N + Q\log^2 N)$。

Steel Slicing 2[4]

给定一个如图所示的多边形。多边形有 N 列，每列向上的长度为 h_i，向下的长度为 l_i。

你可以将多边形切若干刀，每一道只能恰好将一个多边形分成两个。求出需要最少的刀数，使得剩下的多边形都是矩形。



$1 \leq N \leq 250\,000$
$1 \leq h_i, l_i \leq 1\,000\,000$

1024 MB, 1 s

[4]XXI Open Cup. Grand Prix of Korea, Problem L

Steel Slicing 2

全部切成矩形相当于不存在凹的顶点，而每一刀只会让凹顶点数减少 1 或 2，因此问题转化为最大化一刀切掉 2 个凹顶点的刀数。

Steel Slicing 2

全部切成矩形相当于不存在凹的顶点，而每一刀只会让凹顶点数减少 1 或 2，因此问题转化为最大化一刀切掉 2 个凹顶点的刀数。

竖直的切法是平凡的，水平的切法也可以用笛卡尔树预处理出来。如果一种竖直的切法和水平的切法相交，则只能取它们中的一种，因此要求的是二分图的最大独立集。

Steel Slicing 2

全部切成矩形相当于不存在凹的顶点，而每一刀只会让凹顶点数减少 1 或 2，因此问题转化为最大化一刀切掉 2 个凹顶点的刀数。

竖直的切法是平凡的，水平的切法也可以用笛卡尔树预处理出来。如果一种竖直的切法和水平的切法相交，则只能取它们中的一种，因此要求的是二分图的最大独立集。

最大独立集等于点数减去最大匹配，而注意到本题中的图是每种水平的切法向一个区间连边，因此可以从左到右枚举竖直的切法，贪心匹配右端点最小的区间。用堆维护，时间复杂度 $O(N \log N)$。

Interesting Drug[5]

N 件物品排列在数轴上，第 i 件物品的坐标为 i。一个人从某个点开始，可以任意左右移动，当经过一件物品时，他就会取得这件物品 (不能跳过)。若物品 i 是第 C_i 件取走的，则会带来 D_i 的收益。

对每个 $1 \leq i \leq N$，求出初始坐标为 i 时，取得所有物品的最大收益。

$2 \leq N \leq 300\,000$
$1 \leq D_i \leq 10^9$

1024 MB, 1 s

[5] XIX Open Cup. Grand Prix of Korea, Problem K

Interesting Drug

倒过来考虑这个过程，则相当于每次放回最左或最右的被取走的物品。记状态 (L, R) 表示当前被取走的物品区间是 $[L, R]$，则放回物品的过程是从 $(1, N)$ 开始，每一次将 L 增加 1 或将 R 减少 1，直至到达 (i, i)。

Interesting Drug

倒过来考虑这个过程，则相当于每次放回最左或最右的被取走的物品。记状态 (L, R) 表示当前被取走的物品区间是 $[L, R]$，则放回物品的过程是从 $(1, N)$ 开始，每一次将 L 增加 1 或将 R 减少 1，直至到达 (i, i)。

物品 i 带来收益的条件是从 $(i, i + C_i - 1)$ 走到了 $(i + 1, i + C_i - 1)$ 或从 $(i - C_i + 1, i)$ 走到了 $(i - C_i + 1, i - 1)$。将这些点连同 $(1, N)$ 和所有的 (i, i) $(1 \le i \le N)$ 作为关键点进行 DP，用树状数组优化即可。时间复杂度 $O(N \log N)$。

Find the LCA[6]

给定数列 $A_1, A_2, \ldots, A_N$。考虑所有 N 个点的有根树，其中顶点 i $(2 \le i \le N)$ 的父亲为 p_i $(1 \le p_i < i)$，点权为 A_i。

定义一棵树的分数为顶点 $\operatorname{lca}(N-1, N)$ 的子树内所有顶点点权的乘积。求所有 $(N-1)!$ 种树的分数的和，对 $998\,244\,353$ 取模。

$3 \le N \le 250\,000$
$1 \le A_i \le 998\,244\,353$

1024 MB, 7 s

[6] XXI Open Cup. Grand Prix of Tokyo, Problem F

Find the LCA

考虑对于一个包含 $N-1$ 和 N 的点集 S 求出 $\mathrm{lca}(N-1,N)$ 的子树恰好是集合 S 的方案数。若 $1\in S$ 则 S 只能是 $\{1,2,\ldots,N\}$；若 $1\notin S$，则集合外的点只能向集合外连边，子树的根可以任意连边，方案数为 $(N-|S|-1)!\cdot(\min S-1)$。重点在于求出子树内部的方案数，即 N 个点的树，$\mathrm{lca}(N-1,N)=1$ 的方案数。

Find the LCA

考虑对于一个包含 $N-1$ 和 N 的点集 S 求出 $\mathrm{lca}(N-1,N)$ 的子树恰好是集合 S 的方案数。若 $1\in S$ 则 S 只能是 $\{1,2,\dots,N\}$；若 $1\notin S$，则集合外的点只能向集合外连边，子树的根可以任意连边，方案数为 $(N-|S|-1)!\cdot(\min S-1)$。重点在于求出子树内部的方案数，即 N 个点的树，$\mathrm{lca}(N-1,N)=1$ 的方案数。

可以发现当 $N\geq 3$ 时，这样的方案数恰好为 $(N-1)!/2$。这可以利用构造双射来证明：

Find the LCA

考虑对于一个包含 $N-1$ 和 N 的点集 S 求出 $\text{lca}(N-1,N)$ 的子树恰好是集合 S 的方案数。若 $1 \in S$ 则 S 只能是 $\{1,2,\ldots,N\}$；若 $1 \notin S$，则集合外的点只能向集合外连边，子树的根可以任意连边，方案数为 $(N-|S|-1)! \cdot (\min S - 1)$。重点在于求出子树内部的方案数，即 N 个点的树，$\text{lca}(N-1,N)=1$ 的方案数。

可以发现当 $N \geq 3$ 时，这样的方案数恰好为 $(N-1)!/2$。这可以利用构造双射来证明：

- 对于一棵 $\text{lca}(N-1,N)=1$ 的树，设 1 的孩子中子树包含 $N-1$ 的那一个为 x，则取出 x 的子树并插入 1 到 N 的路径，即得到了一棵 $\text{lca}(N-1,N)=x\neq 1$ 的树。
- 对于一棵 $\text{lca}(N-1,N)=x\neq 1$ 的树，取出 x 的子树 (但保留 N 所在的子树) 并接到 1 上，即得到了一棵 $\text{lca}(N-1,N)=1$ 的树。

Find the LCA

考虑对于一个包含 $N-1$ 和 N 的点集 S 求出 $\mathrm{lca}(N-1,N)$ 的子树恰好是集合 S 的方案数。若 $1\in S$ 则 S 只能是 $\{1,2,\ldots,N\}$；若 $1\notin S$，则集合外的点只能向集合外连边，子树的根可以任意连边，方案数为 $(N-|S|-1)!\cdot(\min S-1)$。重点在于求出子树内部的方案数，即 N 个点的树，$\mathrm{lca}(N-1,N)=1$ 的方案数。

可以发现当 $N\geq 3$ 时，这样的方案数恰好为 $(N-1)!/2$。这可以利用构造双射来证明：

- 对于一棵 $\mathrm{lca}(N-1,N)=1$ 的树，设 1 的孩子中子树包含 $N-1$ 的那一个为 x，则取出 x 的子树并插入 1 到 N 的路径，即得到了一棵 $\mathrm{lca}(N-1,N)=x\neq 1$ 的树。
- 对于一棵 $\mathrm{lca}(N-1,N)=x\neq 1$ 的树，取出 x 的子树 (但保留 N 所在的子树) 并接到 1 上，即得到了一棵 $\mathrm{lca}(N-1,N)=1$ 的树。

因此问题转化为了对每个 $1\leq k\leq N-3$ 求 $\sum_{S\subset\{2,3,\ldots,N-2\},|S|=k}(\min S)\cdot\prod_{j\in S}A_j$。这可以用分治 FFT 来解决，时间复杂度为 $O(N\log^2 N)$。

Japanese Knowledge[7]

给定非降正整数序列 $A_1, A_2, \ldots, A_N$。对每个 $0 \le k \le N$，求满足下列条件的非降非负整数序列 $x_1, x_2, \ldots, x_N$ 的数量，对 $998\,244\,353$ 取模。

- $x_i \le A_i$，$1 \le i \le N$。
- 恰好 k 个下标 i 满足 $x_i = A_i$。

$1 \le N \le 250\,000$
$1 \le A_1 \le A_2 \le \cdots \le A_N \le 250\,000$

1024 MB，10 s

[7] XXI Open Cup. Grand Prix of Tokyo, Problem J

Japanese Knowledge

假设没有对满足 $x_i = A_i$ 的下标个数的限制，设答案为 $f(A_1, A_2, \ldots, A_N)$，则 $f(A)$ 相当于从 $(0,0)$ 向上或向右走到 (N, A_N) 且不超出边界的路径数，可以通过分治 FFT 求出：

Japanese Knowledge

假设没有对满足 $x_i = A_i$ 的下标个数的限制，设答案为 $f(A_1, A_2, \ldots, A_N)$，则 $f(A)$ 相当于从 $(0, 0)$ 向上或向右走到 (N, A_N) 且不超出边界的路径数，可以通过分治 FFT 求出：

- 已知右边界的每一个点到 (N, A_N) 的路径数，要求出下边界的每一个点到 (N, A_N) 的路径数。
- 从 $(N/2, A_{N/2})$ 向下和向右画两条线，将区域分成三部分。
- 递归计算右上部分的答案。然后进行右下部分的矩形边界之间的转移，由于系数是组合数，可以用 FFT 来优化转移。最后递归计算左下部分的答案。

不妨设 $A_N = O(N)$，则时间复杂度为 $O(N \log^2 N)$。

Japanese Knowledge

现在考虑求出恰好 k 个下标 i 满足 $x_i = A_i$ 的方案数，发现它等于 $f(A_{k+1}-1, A_{k+2}-1, \ldots, A_N-1)$。这可以通过构造双射来证明：

Japanese Knowledge

现在考虑求出恰好 k 个下标 i 满足 $x_i = A_i$ 的方案数，发现它等于 $f(A_{k+1}-1, A_{k+2}-1, \dots, A_N-1)$。这可以通过构造双射来证明：

- 对于一组原问题的解，去掉所有 $x_i = A_i$ 的位置就能得到一组新问题的解。
- 对于一组新问题的解，从前往后考虑，把每个数 x'_i 放在最小的未被占用的位置 j 满足 $x'_i < A_j$，最后令所有未确定的 x_j 等于 A_j，即得到了一组原问题的解。

在分治 FFT 过程中统计答案，时间复杂度仍为 $O(N\log^2 N)$。

Hit[8]

给定数轴上的 n 条线段 $[l_i, r_i]$，在数轴上放至多 n 个点，使得每条线段至少包含其中的一个点，且一条线段包含点数的最大值最小。

$1 \leq n \leq 10^5$
$-10^9 \leq l_i < r_i \leq 10^9$

512 MB, 2 s

[8] XX Open Cup. Grand Prix of Gomel, Problem H

Hit

先二分答案，转化成判断是否存在一组解，使得一条线段至多包含 k 个点。

Hit

先二分答案，转化成判断是否存在一组解，使得一条线段至多包含 k 个点。

考虑 DP。设 x 是好点当且仅当存在一个放点的方案选择了 x，且所有满足 $r_i \geq x$ 的线段都至少包含一个点，至多包含 k 个点。判断是否有解只需要看 $-\infty$ 是不是好点。

Hit

先二分答案，转化成判断是否存在一组解，使得一条线段至多包含 k 个点。

考虑 DP。设 x 是好点当且仅当存在一个放点的方案选择了 x，且所有满足 $r_i \geq x$ 的线段都至少包含一个点，至多包含 k 个点。判断是否有解只需要看 $-\infty$ 是不是好点。

记 $\mathrm{next}(x)$ 为最小的好点 $y > x$ 满足不存在一条线段 $[l_i, r_i]$ 被 (x, y) 严格包含。若选择了 x，则之后选择的点一定是 $\mathrm{next}(x), \mathrm{next}(\mathrm{next}(x)), \ldots$。若 x 和 $\mathrm{next}^k(x)$ 被一条线段包含，则这条线段包含至少 $k+1$ 个点，因此 x 不是好点；否则，x 是好点。

Hit

先二分答案，转化成判断是否存在一组解，使得一条线段至多包含 k 个点。

考虑 DP。设 x 是好点当且仅当存在一个放点的方案选择了 x，且所有满足 $r_i \geq x$ 的线段都至少包含一个点，至多包含 k 个点。判断是否有解只需要看 $-\infty$ 是不是好点。

记 $\mathrm{next}(x)$ 为最小的好点 $y > x$ 满足不存在一条线段 $[l_i, r_i]$ 被 (x, y) 严格包含。若选择了 x，则之后选择的点一定是 $\mathrm{next}(x), \mathrm{next}(\mathrm{next}(x)), \ldots$。若 x 和 $\mathrm{next}^k(x)$ 被一条线段包含，则这条线段包含至少 $k+1$ 个点，因此 x 不是好点；否则，x 是好点。

将坐标离散化后，用倍增优化找 $\mathrm{next}^k(x)$ 的过程，则检验一次的复杂度为 $O(n \log n)$，总时间复杂度 $O(n \log^2 n)$。

Hit

先二分答案，转化成判断是否存在一组解，使得一条线段至多包含 k 个点。

考虑 DP。设 x 是好点当且仅当存在一个放点的方案选择了 x，且所有满足 $r_i \geq x$ 的线段都至少包含一个点，至多包含 k 个点。判断是否有解只需要看 $-\infty$ 是不是好点。

记 $\text{next}(x)$ 为最小的好点 $y > x$ 满足不存在一条线段 $[l_i, r_i]$ 被 (x, y) 严格包含。若选择了 x，则之后选择的点一定是 $\text{next}(x), \text{next}(\text{next}(x)), \ldots$。若 x 和 $\text{next}^k(x)$ 被一条线段包含，则这条线段包含至少 $k+1$ 个点，因此 x 不是好点；否则，x 是好点。

将坐标离散化后，用倍增优化找 $\text{next}^k(x)$ 的过程，则检验一次的复杂度为 $O(n \log n)$，总时间复杂度 $O(n \log^2 n)$。

事实上，可以先贪心构造一组解，即每次取还未包含点的线段中最小的右端点处放一个点。设贪心求出来的解中，一条线段最多包含了 t 个点，则意味着存在一条长线段包含了 $t-1$ 条互不相交的短线段，因此答案至少为 $t-1$。因此只需要检查 $t-1$ 而不需要二分，时间复杂度变为 $O(n \log n)$。

Chess Tournament[9]

在大赛中 n 位选手要两两进行一场比赛。大赛包含若干轮，每一轮至多有 k 对选手同时比赛 (因此这至多 $2k$ 位选手必须互不相同)。设计一个比赛方案使得总轮数最少。

$2 \le n \le 200$
$1 \le k \le n/2$

512 MB, 2 s

[9] XXI Open Cup. Grand Prix of Samara, Problem I

Chess Tournament

不妨先考虑 $k = \lfloor n/2 \rfloor$ 的情况。

Chess Tournament

不妨先考虑 $k = \lfloor n/2 \rfloor$ 的情况。

若 n 是奇数，则可以第 i 轮让每一对与 i 距离相同的人比赛，例如第一轮的比赛为 $(2, n), (3, n-1), (4, n-2), \ldots, (\frac{n+1}{2}, \frac{n+3}{2})$。若 n 是偶数，则在 $n-1$ 的方案的基础上，第 i 轮增加 (i, n) 即可。

Chess Tournament

不妨先考虑 $k=\lfloor n/2\rfloor$ 的情况。

若 n 是奇数，则可以第 i 轮让每一对与 i 距离相同的人比赛，例如第一轮的比赛为 $(2,n),(3,n-1),(4,n-2),\ldots,(\frac{n+1}{2},\frac{n+3}{2})$。若 n 是偶数，则在 $n-1$ 的方案的基础上，第 i 轮增加 (i,n) 即可。

考虑一般情况。将 $k=\lfloor n/2\rfloor$ 时每一轮的比赛按顺序写下，n 是偶数时 (i,n) 在最前面。例如 $n=6$ 时应写下：

Chess Tournament

不妨先考虑 $k = \lfloor n/2 \rfloor$ 的情况。

若 n 是奇数，则可以第 i 轮让每一对与 i 距离相同的人比赛，例如第一轮的比赛为 $(2,n),(3,n-1),(4,n-2),\dots,(\frac{n+1}{2},\frac{n+3}{2})$。若 n 是偶数，则在 $n-1$ 的方案的基础上，第 i 轮增加 (i,n) 即可。

考虑一般情况。将 $k = \lfloor n/2 \rfloor$ 时每一轮的比赛按顺序写下，n 是偶数时 (i,n) 在最前面。例如 $n=6$ 时应写下：

- $(1,6),(2,5),(3,4)$
- $(2,6),(3,1),(4,5)$
- $(3,6),(4,2),(5,1)$
- $(4,6),(5,3),(1,2)$
- $(5,6),(1,4),(2,3)$

Chess Tournament

不妨先考虑 $k=\lfloor n/2\rfloor$ 的情况。

若 n 是奇数，则可以第 i 轮让每一对与 i 距离相同的人比赛，例如第一轮的比赛为 $(2,n),(3,n-1),(4,n-2),\ldots,(\frac{n+1}{2},\frac{n+3}{2})$。若 n 是偶数，则在 $n-1$ 的方案的基础上，第 i 轮增加 (i,n) 即可。

考虑一般情况。将 $k=\lfloor n/2\rfloor$ 时每一轮的比赛按顺序写下，n 是偶数时 (i,n) 在最前面。例如 $n=6$ 时应写下：

- $(1,6),(2,5),(3,4)$
- $(2,6),(3,1),(4,5)$
- $(3,6),(4,2),(5,1)$
- $(4,6),(5,3),(1,2)$
- $(5,6),(1,4),(2,3)$

注意到每个人在相邻两轮中的位置至多相差 1。将每一行按顺序拼接起来，则同一个人不会在连续 $\lfloor n/2\rfloor-1$ 场比赛中出现两次，因此只需要每连续 k 场比赛为一轮，显然是最优的。

谢谢大家！