Petrozavodsk Summer 2024 Day 5 — Problem A: Building Marble Tracks

1 题目大意

原题链接：QOJ 题面

给定 n 条直线轨道，每条为线段 (x_1, y_1)–(x_2, y_2)，且 $x_1 \neq x_2$。按输入顺序（全局编号 $1, \ldots, n$）依次尝试固定：当且仅当当前线段与此前已经固定的任意线段在内部没有相交（端点处接触不算），就把它固定；否则跳过。

下文若未加限定，“相交/不相交”均指“严格内部相交/不相交”。

输入 第一行一个整数 n。接下来 n 行，每行四个整数 x_1, y_1, x_2, y_2 表示一条线段（保证 $x_1 \neq x_2$）。

输出 第一行输出被保留的数量 k。第二行输出 k 个编号，按升序排列。

2 数据范围

- $1 \le n \le 6 \times 10^4$；
- 坐标范围 $-10^5 \le x_1, y_1, x_2, y_2 \le 10^5$，且每条线段满足 $x_1 \neq x_2$；
- 任意两条不同线段至多在一个点相交（不存在重叠成段）；允许多条线段在同一点相交；
- 时间限制 4 秒，内存限制 256 MB。

3 详细做法

朴素做法在第 i 条线段时与所有已保留线段逐一判交，最坏 $\mathcal{O}(n^2)$。正确做法采用

分块 ＋ 线段扫描（Bentley-Ottmann） ＋ 块内筛选

可把复杂度降至 $\mathcal{O}\big(n^{\frac{3}{2}} \log n\big)$。

分块 设块大小 $B = \lceil \sqrt{n} \rceil$，按输入顺序把 n 条线段分成 $\lceil \frac{n}{B} \rceil$ 个连续块。

全局编号固定为输入序号并在全流程中使用；块内任何引用均以该全局编号为准。

处理到某一块 T 时，记此前已保留下来的集合为 S。初始 $S = \varnothing$；每次仅在与已入选线段不相交时才加入（见 Step 2 与 Step 3）。据此可得：

- 基例：此时 $S = \varnothing$；

- 归纳步：若当前 S 内两两不相交，且仅当待加入线段与 S 中任意线段不相交时才加入，则加入后任一对线段仍不相交。

因此在任意时刻 S 内部两两不相交。

三步处理

Step 1： **处理块内相交**：对 T 中至多 B 条线段两两做一次相交测试，记录布尔关系 $I(i,j)$ 表示两个线段是否相交。

Step 2： **跨集合相交**：在 $S \cup T$ 上做一次线段扫描（Bentley-Ottmann）。只要发现某条 $t \in T$ 与某条 $s \in S$ 相交，立刻把 t 标记为无效并从状态结构删除，其后续相关事件失效（惰性删除），并对 t 删除后在状态结构中新形成的一对原上下邻居立即执行一次相邻对检查。

Step 3： **按顺序筛选**：按全局编号从小到大遍历 T。若该线段被 Step 2 标记为无效则跳过；否则与本块中已加入且更早的线段按 $I(u,v)$ 检查，若存在相交则跳过，否则把它加入 S。

4 Bentley-Ottmann 流程

对 x 做扫描线，扫描线自左向右移动，同时维护两个结构：事件队列与状态结构。

端点 对每条线段，以 x 坐标升序定义左端点与右端点；若输入给出 $x_1 > x_2$，则先交换两端点（题目保证 $x_1 \neq x_2$）。在某一固定的 $x = x_0$ 上批处理事件时，状态结构中的相对高低与相邻关系均按当前扫描位置的右极限 $x \to x_0^+$ 的截线 y 值来判定，即先按照当前扫描位置 x 的截线 y 值比较，相同时再按照斜率比较。

事件 按 x 坐标从小到大处理，事件分为：

- START：线段的加入事件（左端点）；
- END：线段的删除事件（右端点）；
- CROSS：线段的相交事件（仅包含 T–T 的相交事件）。

初始化时加入 $S \cup T$ 的所有 START 与 END 事件。扫描过程中，**每当任意一对相邻线段形成或变更相邻关系时，先进行一次相邻对交点检查：**

- 若为 S–T，且判定相交，立即从状态结构删除 T 中该线段并将其后续相关事件视为失效（惰性删除），且对其原上下邻居立即执行一次相邻对检查；
- 若为 T–T，则计算并将其唯一的交点（若存在且在当前批次 x 的右侧）作为 CROSS 事件加入队列；若交点恰在当前批次 $x = x_0$，归入该批次的 CROSS 批事件；
- 若为 S–S，由归纳它们不相交，无需处理。

状态 用平衡树维护当前被扫描线穿过的所有活跃线段，按它们在当前扫描位置右极限 $x \to x_0^+$ 的截线 y 值从下到上有序（即先按照当前扫描位置 x 的截线 y 值比较，相同时再按照斜率比较）。处理事件时，若事件关联线段已不在状态结构，则跳过该事件（惰性删除原则）。

同批次处理顺序 在同一 $x = x_0$ 上按固定顺序批处理：

$$\text{END} \;\to\; \text{CROSS} \;\to\; \text{START}.$$

同类事件内部次序如下：

- END：按当前状态从下到上删除。每删除一条线段，立即检查其原上下邻居是否成为相邻，并执行相邻对检查。
- CROSS：对每一个交点 p，取穿过 p 的所有当前活跃的 T 内线段，按它们在 $x \to x_0^-$ 时由下到上的顺序整体取反序（例如下到上的序列 $1, 2, 3$ 变为 $3, 2, 1$）。完成反序后，分别对该组与其外部上下邻居形成的新相邻对执行一次相邻对检查。由于先处理 END，其右端点在 p 的线段已不活跃；由于后处理 START，其左端点在 p 的线段尚未加入，故“当前活跃”恰对应“内部穿过 p”。
- START：将线段插入状态结构；分别与其上下邻居做相邻对检查。若与某个邻居属于 S–T 相交则即时删除两条线段中的 T 线段并对其原上下邻居继续检查；若与某个邻居属于 T–T 相交，则将对应 CROSS 入队。

无论在何处触发 S–T 的即时删除，均立即从状态结构移除并对其原上下邻居形成的新相邻对执行一次相邻对检查。

5 正确性证明

目标：在扫描线过程中仅检查状态结构中的相邻对即可发现 $S \cup T$ 内全部内部交点；据此可知 Step 2 的跨集合过滤与 Step 3 的顺序筛选与题意一致。记当前批处理坐标为 x_0。状态结构维护穿过 $x = x_0$ 的活跃线段，按 $x \to x_0^+$ 的截线 y 值自下而上有序。同一 $x = x_0$ 的事件按 END $\to$ CROSS $\to$ START 批处理：在 CROSS 批中活跃的线段恰为内部穿过该交点的线段。

引理 1 (左侧最早交点的连续块)**.** *设 $x^\star$ 为任意两线段内部首次相交的 x 坐标，取 $p = (x^\star, y^\star)$。令 G 为在 p 处内部经过 p 的全部活跃线段集合。则在 $x = x^\star - \varepsilon$ 的状态序列中，G 形成一个连续块。*

证明. 反证。若存在 $a, b \in G$ 与 $c \notin G$，在 $x^\star - \varepsilon$ 的序列中 c 位于 a, b 之间，则当 x 增至 $x^\star$ 时，c 必通过端点或内部相交离开该区间。端点事件与“c 位于二者之间”矛盾；内部相交则给出 $x < x^\star$ 的更早相交，违背 $x^\star$ 的最早性。故 G 必为连续块。 □

引理 2 (相邻对检查的完备性). *在扫描中，只要每当相邻关系创建或变更时立即检查该相邻对，即可发现全部内部交点（含同点多线相交），且不会报告端点接触。*

证明. 情形一：两线在 p 简单相交。最早相交点的左侧，两线相邻；相邻关系首次出现或变更即被检查并命中。

情形二：多线在同一点 p 内部相交。由引理 1，G 在 $x^\star - \varepsilon$ 为连续块，跨过 $x^\star$ 后块内上下次序整体取反。对 G 内任意一对在 p 相交的相邻的线段，必在某一步成为相邻而被检查命中。由于 END 与 START 已先后移除/未插入端点在线的线段，端点接触不会被报告。 □

引理 3 (跨集合过滤). *在 Step 2 中，对任意 $t \in T$，当且仅当存在 $s \in S$ 与之内部相交时，t 被即时删除。*

证明. 充分性：由引理 2，s 与 t 在相邻关系创建或变更时被检查并命中，算法立即删除 t 并继续相邻检查。必要性：仅当相邻对判定为内部相交时才删除 t；对 S–S 不作删除，故无伪阳性。 □

引理 4 (块内顺序筛选). *完成 Step 2 后，T 中剩余线段与 S 互不相交。记 T' 为剩余集合，则按全局编号递增处理 T'，仅与已入选且编号更小的同块线段比较，并依据预计算的 $I(u, v)$ 决定取舍，与题意等价。*

定理 1 (正确性). *由引理 2 与引理 3，Step 2 通过相邻对检查完整过滤掉与 S 相交的 T 内线段（含同点多线相交）。由引理 4，Step 3 的块内顺序筛选与题意中的按输入顺序取舍一致。故算法正确。*

6 复杂度分析

令本块大小 $|T| \le B$，本块内 T–T 的相交点数为 q。一次扫描产生的队列事件数为

$$E \;=\; 2\big(|S| + |T|\big) \;+\; q,$$

其中 $2(|S| + |T|)$ 来自所有 START/END，q 来自所有 CROSS。在扫描过程中，每个事件只触发 $\mathcal{O}(1)$ 组相邻检查；此外，每条 $t \in T$ 至多一次被 S 命中并即时删除，该操作在状态结构内引起的额外相邻检查次数为 $\mathcal{O}(1)$，因而本块扫描时间为

$$\mathcal{O}\big((E + |T|) \cdot \log(|S| + |T|)\big) = \mathcal{O}\big((|S| + q) \log n\big).$$

由“至多一个交点、且不重叠”可知 $q = \mathcal{O}(B^2)$；Step 1 与 Step 3 各为 $\mathcal{O}(B^2)$。于是本块总时间

$$\mathcal{O}\big((|S| + B^2) \log n + B^2\big) = \mathcal{O}\big((|S| + B^2) \log n\big).$$

将块按处理顺序编号为 $1, \dots, m$，其中 $m = \lceil \frac{n}{B} \rceil$，第 b 块开始时有 $|S| \le (b-1)B$，求和得

$$\sum_{b=1}^{m} \mathcal{O}\big(((b-1)B + B^2) \log n\big) = \mathcal{O}\big((m^2 B + B^2 m) \log n\big) = \mathcal{O}\big(n^{\frac{3}{2}} \log n\big),$$

空间为 $\mathcal{O}(n)$。

7 参考资料

1. J. L. Bentley and T. Ottmann, *Algorithms for Reporting and Counting Geometric Intersections*, IEEE Transactions on Computers, 1979.

2. Mark de Berg, Otfried Cheong, Marc van Kreveld, and Mark Overmars, *Computational Geometry: Algorithms and Applications*, 3rd ed., Springer.

3. Wikipedia, *Bentley-Ottmann algorithm*, https://en.wikipedia.org/wiki/Bentley%E2%80%93Ottmann_algorithm.