《Defend City》命题报告

徐哲安

November 28, 2020

1 试题来源

本题是笔者为 2020 年 CCPC 秦皇岛站命制的一道题[1]。

2 题目大意

平面上有 n 个点 (x_i, y_i, d_i)，根据 d_i 的不同有四种类型，每个点 i 覆盖一个四分之一平面，方向分别为右上、左上、左下、右下（对应 $d_i = 1, 2, 3, 4$）。

例如，若 $d_i = 1$，点 i 能够覆盖的区域为**右上** $\{(a, b) \mid a \geq x_i, b \geq y_i\}$。

请你选择数量最少的点，使得任意 $(x, y) \in \mathbb{R}^2$ 都能被至少一个点覆盖。

3 数据范围

对于 10% 的测试数据，满足 $n \leq 20$, $\sum n \leq 100$。

对于 30% 的测试数据，满足 $n \leq 100$, $\sum n \leq 500$。

对于 40% 的测试数据，满足 $n \leq 1000$, $\sum n \leq 5000$。

对于 70% 的测试数据，满足 $n \leq 10^5$, $\sum n \leq 5 \times 10^5$。

$1 \leq T \leq 10^4$, $1 \leq n \leq 10^6$, $\sum n \leq 5 \times 10^6$, $1 \leq x_i, y_i \leq n$, $1 \leq d_i \leq 4$。

时间限制：2 s

空间限制：512 MB

4 解题过程

4.1 初步分析

显然四个方向都至少需要一个点才能覆盖整个平面，同时我们记四个方向的点集为 P_1, P_2, P_3, P_4。对于每个点集，显然只需要留下不被点集内其他点覆盖的点，剩下所有点的 y 随 x 增大而单调变化。设每个方向选择的子集 $Q_i \subseteq P_i$。设 Q_i 覆盖区域的并为 S_i，设 P_i 覆盖区域的并为 T_i。

[1]参见：https://codeforces.com/gym/102769/problem/D

4.2 关键性质

通过观察一些简单的情形，我们可以发现如下结论：

定理 1. *存在最优的解，使得 $|Q_1| = |Q_3| = 1$ 或者 $|Q_2| = |Q_4| = 1$。*

证明. 任取一个最优的解 $\{Q_1, Q_2, Q_3, Q_4\}$，不妨设 $Q_1 \geq 2$。

任取 Q_1 中一对相邻的点 p, q，观察它们覆盖区域的边界交点 r。如果 r 被 Q_2 或者 Q_4 中的点覆盖，显然 Q_1 中删除一个点仍然能覆盖整个平面，与前提矛盾。那么 Q_1 中所有覆盖区域的交点 r_i 都只能被 Q_3 中的点覆盖。反过来，Q_3 和 Q_1 同理。

之后考虑 Q_1, Q_3 最左侧覆盖区域的边界交点 t，若 t 被 Q_2 中的一个点覆盖，显然左上的所有未覆盖区域都能被这个点覆盖，所以 Q_2 可以只保留一个点，Q_4 同理。即 $|Q_2| = |Q_4| = 1$。 □

考察如下示例，可以更直观地理解结论及其证明：

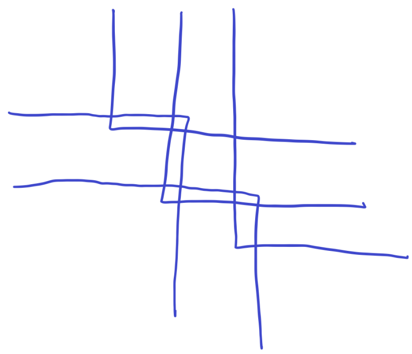

图 1: 只考虑 Q_1 和 Q_3 的覆盖区域，最优解一定形如上图所示

4.3 进一步分析

接下来不妨令 $|Q_2| = |Q_4| = 1$，容易发现此时 S_1 与 S_3 的轮廓线的交点要落入 T_2 与 T_4 之中，且两个交点之间 S_3 轮廓线始终在 S_1 轮廓线右方和上方。

不妨枚举 Q_1 最上方的点，我们接下来选择 Q_3 中最上方的点（根据单调性，同时也是最左侧的点）。显然，我们需要在满足左侧被完整覆盖的情况下，选择的点尽量往右（否则可以被更优的解代替），也即我们贪心取最优的 Q_3 最上方的点，这个后继是唯一的。同样这个 Q_3 中的点也有 Q_1 中的唯一后继。不断选择唯一后继，直到在 T_4 中有交点。我们就得到了以 Q_1 中的某个点开始的最优解。由于我们作了一些方便解决问题的假设，在实现算法的时候，还需要旋转整个平面使得所有情况都被考虑到。

如果我们暴力 $O(n)$ 枚举所有起始点，暴力枚举 $O(n)$ 步，同时暴力 $O(n)$ 找到下一个点，时间复杂度为 $O(n^3)$，可以得到 30 分。

一个简单的优化是发现每个点的后继是唯一的，我们可以先 $O(n^2)$ 预处理每个点的后继，这样就能将时间复杂度优化到 $O(n^2)$，可以得到 40 分。

4.4 倍增优化

实际上，“每个点的后继唯一”是一个非常有用的性质，我们可以通过**倍增**将暴力模拟的时间复杂度进行优化。

具体来说，利用二分查找可以 $O(\log n)$ 找出每个点的后继，同时，对于任意一个点 i，预处理跳 2^j 步后会到达的点 $f[i][j]$。我们就可以将暴力找下一个节点的过程用倍增替代，只需要 $O(\log n)$ 的时间复杂度就能完成整个模拟。

总时间复杂度为 $O(n\log n)$，可以得到 70 分。

值得注意的是，每个点对应后继形成了**基环内向树森林**的结构。由此，我们可以利用值域小的特点 $O(n)$ 预处理每个点的后继，并利用**记忆化搜索**替代倍增，得到一个常数较大，实现较复杂的 $O(n)$ 做法。

此外，孔朝哲学长也提出一个基于数组+动态规划的简洁 $O(n)$ 做法[2]，这里不详细展开。

4.5 最终解法

我们可以证明，当答案 ≥ 5 时，最优解中 Q_1 最上方的点 p_k 是能被唯一确定的。

我们需要 Q_1 中的第二个点尽量往下，这就需要 Q_3 中的第一个点尽量往右。可以发现 T_1 与 T_2 轮廓线交点，左侧或者下方的点 p_k（根据相交的方向不同）一定是最优的选择。

对于 P_1 中 y 更大的点 p_t ，显然新增的覆盖区域是 p_k 新增覆盖区域的子集。对于 y 更小的点 p_t，会使得对应在 P_3 中的后继 p_s 更劣。之后 p_s 在 P_1 中的后继就与 p_k 无关了。

一旦我们确定了最优解中 Q_1 的第一个点 p_k，我们就只需要模拟 $O(n)$ 步就可以了。

对于 $O(1)$ 寻找后继，判断一个点是否在 T_i 中可以用数组 $O(n)$ 预处理完成。具体来说，一个点会对前缀 x 或者后缀 x 产生贡献，我们在对应的位置处打标记，最后做一个前缀后缀 min 或者 max 就可以了。

综上，我们得到了一个 $O(n)$ 的简洁做法。

5 命题背景

在命制本题之前，笔者曾经提出了这样一个问题：给定一个目标矩形和 n 个坐标固定的可用矩形，请选择数量最少的可用矩形覆盖目标矩形。笔者通过一段时间的思考，未能得到任何一个多项式时间复杂度做法（有可能本身不存

[2]提交链接：https://codeforces.com/gym/102769/submission/97104014

在）。于是，笔者在这个问题上增加一个**特殊化**的限制：任意一个矩形都是大小相同的正方形。之后稍加修改，就得到现在这道试题。

将问题特殊化后，带来了许多意想不到的有趣的性质，尤其是各种单调性都能被巧妙地运用。这启发我们，在命制的试题不尽如人意的时候，可以考虑稍加修改题目条件，如**一般化**等。对于一些不存在多项式做法的题目，可以反过来考虑**特殊化**，由此往往可以带来更多有用、有趣的性质，得到一道更好的试题。

6 总结

在 CCPC 现场赛中，没有队伍通过本题。截至目前，在 Gym 上仅有东京大学的一支队伍虚拟参加通过本题 [3]。笔者认为，本题难度接近 NOI 中的第二题，属于中档题。

本题最终的解法非常简洁而优美，代码量较小，同时所用算法没有超过联赛范围，主要考察了选手的思维能力与问题分析能力。与从算法出发的应用题、从原有试题出发的改编题不同，本题从问题出发，使得最终的题面与解法非常自然，同时具有趣味性与创新性。这也启发我们在命制试题的时候，更多从问题而不是解法出发，往往能够得到更加令人耳目一新的好题。

7 致谢

感谢周航锐同学与我讨论本题的算法。

8 参考资料

笔者的知乎回答

[3]提交链接：https://codeforces.com/gym/102769/submission/99776965