Codechef FEB 13 Efficient Painting 解题报告

金策

October 26, 2015

1 题意简述

你有一块 $n \times n$ 的正方形的画布，初始时每个格子都是白色的。你每次可以选择一个子矩形并对它进行操作，矩形的边长是整数，且边必须平行于坐标轴（矩形的边界必须沿着格子线画）。有下面三种可选用的操作：

- White - 矩形内全涂成白色。
- Black - 矩形内全涂成黑色。
- Flip - 矩形内的白色变成黑色，黑色变成白色。

你会拿到一个所要求的最终图案。你需要用尽量少的操作次数，将画布上的图案变成所要求的样子。你并不需要求出最优解，但是你用的次数越少则获得的分数越多。

数据生成与评分方式如下：$10 \le n \le 50$ 共有 50 个数据文件，每个都是这样生成的：一个整数 n 从 $[10, 50]$ 中均匀随机抽取。一个实数 p 从 $[0.4, 0.6]$ 中均匀随机抽取。然后每个格子都是独立的以 p 的概率填为黑色，以 $1-p$ 的概率为白色。

你在每个输入文件的得分是 $10L/n^2$。你的总分是你在每个文件的分数平均值，你的目标是最小化你的分数。

2 题目解答

2.1 简单粗暴的做法

考虑全用 B 操作。最终图案里哪些格子是黑的，就用 B 操作把这个格子涂黑。

这个方法是可以改进的，因为黑格子可能有相当一部分是连起来的，那么每一次可以把连着的两个格子一起涂黑。

或者我们每次考虑当前所需要的最大黑色子矩形并涂黑。如果在有些大块的黑色矩形中出现了白色坏点，可以先全涂黑再把白的挖掉。然而数据的生成方式决定了产生的连通矩形往往不会很大，所以这个优化并不明显。

如果所需要的黑色比较多（当 $p > 0.5$) 时，这样做的次数也相应的会变多。这时可以考虑把黑白反过来（对整个盘面做一次 B 操作)，黑色数目就变少了，然后再按刚才的做法做就行了。

2.2 利用翻转操作的做法

刚才的做法仅利用了 B 和 W 操作。现在考虑一下用 F 操作的做法。

考虑所有 $(n+1)^2$ 个格点，一个格点周围有 4 个格子（边界的只有两个，四角的只有一个）。给这个格点赋一个权值，为周围格子中黑色的数量的奇偶性。

现在研究对一个矩形进行 F 操作对格点的权值所产生的影响：对于矩形内部的格点，有 4 个格子的颜色反转，所以黑格数量奇偶性不变；对于矩形边界上的格点，有 2 个格子的颜色反转，所以黑格数量奇偶性也不变；对于四角上的格点，只有 1 个格子颜色反转，所以奇偶性也会反转。也就是说，一次 F 操作可以将矩形四个顶点的权值反转。

我们计算出终盘上各个格点的权值，考虑倒过来操作将其中的 1 全部变为 0，得到全空白的画布。那么一次 F 操作至多可以消去 4 个 1；如果无法同时消去 4 个 1，就消去 3 个 1，添回一个 1。

于是我们要寻找盘面上四个顶点都是 1 的矩形。这可以做到每次查找 $O(n^3)$：对第 i 行，枚举 $j<k$，使得 $w[i][j]=w[i][k]=1$，并将 (j,k) 标记为可行；如果对另外一行也有 (j,k) 可行，就找到了这样的矩形。

如果找不到，就找顶点有三个 1 和一个 0 的矩形，方法也和上面类似。

容易证明变为全 0 之前，总是能找到顶点至少含三个 1 的矩形：因为每行、每列的权值之和都必须是偶数，如果有一个点是 1，那么它所在的行和列都至少有另外的 1，那么这三个 1 所在的矩形一定就合法了。

又由于每一次操作后盘面上 1 的数量或者减少 4，或者减少 2。所以这个贪心方法一定能求得一个合法解。

另外我们可以随机做多次然后取最优，于是在 tsinsen 上就可以通过这题了。

2.3 其他优化

为了取得更高的分数，我们可以将 F 操作和 BW 操作结合起来。还是考虑刚才定义的权值，如果当前的一次 B（或 W）操作能使得权值总和的减少量超过 4，那么就采用这次操作。

也可以采用一些搜索的方法，并将盘面的估价函数定义为 (已经用的次数 + 剩余的权值和/4)。

另外也可以对大范围进行贪心之后，对一些小地方使用状态压缩 DP 取得更好的解。

2.4 总结

这个题是一道实现相对简单，也比较容易取得高分的 challenge 题。