

Ambidextrous Chefs 解题报告

赣州三中 赖金霖

1. 题目大意

M 个人, N 种工具, 每个人使用两种工具, 一个队伍需要满足每种工具至少有一个人使用, 在组成尽量多队伍的情况下要求聘用的人最少。

数据范围, $T=10$, $10 \leq N \leq 100$, $200 \leq M \leq 1000$, 随机生成。

2. 题目解答

这是一道 Challenge 题, 分数为 $(C*N-H)/M$ 的平均数, C 表示完整的队伍数, N 表示工具种数, H 是聘用的人数量, M 是人总数。由数据生成方法可知每种工具出现次数较为平均。

2.1 题目解读与转化

每组成一支队伍, 就能获得 $(N-R)/M$ 的分数, R 表示此队伍聘用的人数量, 所以, 队伍越多得分越高, 同时, 聘用的人越少, 分数越高。一支队伍得分最高时, 每个工具只有一个人使用, 得分为 R/M 。在所有队伍得分最高的情况下, 所有人都被聘用时, 总得分最高, 为 1。

考虑建图, 每种工具视为一个点, 每个人视为连接两个点的一条边。那么一支队伍相当于一个覆盖所有点的边集。我们要求队伍多, 并且用人少, 即边集数量多, 且边集中边数量少。

2.2 做法一

随机选择边加入当前边集, 直到覆盖所有点, 再从空集重新开始, 这种做法非常低效, 在 tsinsen 上得分很低 (8 次方大法好)。

2.3 做法二

考虑贪心, 每次尽量选择两端点均未被覆盖的边加入边集, 再查缺补漏, 给未被覆盖的点随机找一条边加入边集。我们可以随机边的顺序, 多次重复, 取得分最高的那一次。这样做, 在 Codechef 上能获得 0.9 以上的分, 在 tsinsen 上能得到 50 分以上。

2.4 做法三

做法二的第一步显然是一般图的匹配问题, 我们可以使用经典的带花树算法

来做, 这样能保证边集中边尽量少。做到这个优化, 在 Codechef 上就能获得 0.99 以上的分, 在 tsinsen 上至少能得 90 分。

2.5 细节问题

做法二的第二步显然也太暴力了。一个更明智的做法是选择剩余选择机会最多的点和当前点配对, 因为我们还要考虑后来的边集。加上这个细节, 这题就有很大概率跑出最优解了。

3. 复杂度

带花树算法复杂度为 $O(NM)$, 总共约进行 M/N 次, 时间复杂度为 $O(M^2)$, 实际上时限内能随机 40 次。

空间复杂度 $O(N+M)$ 。