

《交朋友》 解题报告

中国人民大学附属中学 许庭强

2021 年 9 月

1 题目

1.1 题目大意

这是一道提交答案题。

有 n 个人，在他们之间组织了若干次聚会。一开始任意两个人都不是朋友。在每一次聚会中，参加的人两两都会成为朋友。现在你忘记了一共组织了多少次聚会，也忘记了每次聚会有谁参加，只知道最后有哪些人成为了朋友。你想要还原出聚会的过程。显然聚会的方案不是唯一的，你想要找出一种使得聚会次数尽量少的方案。

简化版题意：给出一张无向图（是朋友即为有边相连），找到一个包含尽量少的团（聚会）的集合使得这些团并起来是这张图。

1.2 输入格式

第一行包含两个正整数 n, m ，表示人数和朋友对数。

接下来 m 行，每行包含两个数 x, y ，表示 x 和 y 这两个人是朋友。

1.3 输出格式

第一行一个正整数 k ，表示你的方案中聚会的次数。

接下来 k 行，每行表示一次聚会。先输出一个数 r ，表示这次聚会参加的人数。一行内紧接着包含 r 个互不相同的正整数，表示这次聚会参加的人的集合。

你需要保证这 k 个聚会组织后满足输入的 m 对人是朋友，且剩下的人之间没有成为朋友。

你还需要保证 $k \leq m$ 且每次参加聚会的人数之和不超过 $2m$ 。

1.4 数据范围

测试点标号	$n =$	$p =$	特殊性质
1	6	$\frac{1}{2}$	
2	10	$\frac{1}{2}$	
3	50	不存在	A
4	100	$\frac{1}{3}$	
5	100	$\frac{1}{2}$	
6	500	$\frac{1}{5}$	
7	500	$\frac{1}{2}$	
8	1000	$\frac{1}{5}$	
9	1000	$\frac{1}{3}$	
10	1000	$\frac{1}{2}$	

特殊性质 A：可以把人分为两组，使得组内人与人之间都不是朋友。

如果 p 有值，那么说明这一个测试点中，对于每两个人 $i, j (i < j)$ ， i 和 j 有 p 的概率是朋友，有 $1 - p$ 的概率不是朋友。

1.5 评分标准

如果你的输出不符合题目要求，那么得分为 0。

如果你的输出是正确的，令 Z 为你的方案中团的个数，对于这个测试点，你的得分以如下方式计算：

- 如果 $Z \leq Z_0$ ，得分为 S 。
- 如果 $Z > Z_0$ ，得分为 $S \times (\frac{Z_0}{Z})^3$ 。

每个测试点的参数 Z_0 和分数 S 如下：

测试点标号	1	2	3	4	5	6	7	8	9	10
Z_0	4	9	321	288	208	3935	2621	12386	11198	8486
S	4	8	6	9	9	11	11	13	13	16

2 解题过程

2.1 算法 1

通过手动构造可以构造出前两个点的最佳方案。

可以通过前两个测试点得到 12 分。

2.2 算法 2

注意到二分图中不可能存在一个大于等于 3 个点的子图是团，所以对于第 3 个点，只能对每条边新增加一个团，最优解中团的个数就是边的条数。

与算法 1 结合可以通过前三个测试点得到 18 分。

2.3 算法 3

搜索出图中的所有团，然后每次贪心找到覆盖边数最多的团，直到所有边都被覆盖。但此算法在后 6 个点中不能使用，因为无法搜索出所有团。

与前两个算法结合可以得到约 28 分。

2.4 算法 4

我们称一个团的集合 A 为一个状态。对于一个状态 A ，需要满足其中的所有团中的边均在图中出现，即不会覆盖多余的边。

定义 $E(A)$ 为被 A 中的团覆盖的边集， $E'(A)$ 为原图中未被覆盖的边集。

如果 $|E'(A)| > 0$ ，那么我们称 A 为一个“不满”的状态。

如果 $|E'(A)| = 0$ ，那么我们称 A 为一个“满”的状态，即为一个合法的答案。

设定一个估值函数 $f(A) = |A| + |E'(A)|$ ，我们期望找到一个 A 使得 $f(A)$ 的值尽量小。注意到如果 A 为一个“不满”的状态，那么将每条未覆盖的边变成一个团加入后 $f(A)$ 不变。所以一定存在一个“满”的状态使得估值函数的值最小。

维护一个当前状态 A ，初始为空集，并反复执行以下操作：

1. 依次查看每个点 x ，并对于每个当前 A 中不包含 x 的团 C ，查看把 x 加入 C 中并删除 C 中所有与 x 不相邻的点后， $f(A)$ 的值是否变小。如果变小，则执行此操作；如果不变，则有 $\frac{1}{2}$ 的概率执行；否则不执行。
2. 随机构造一个点的排列，维护一个团，初始为空，依次扫每个点并尝试把当前点加入团。设扫过所有点后形成的团为 C 。那么如果把 C 加入 A 中后 $f(A)$ 变小，则将其加入。
3. 查看每个 A 中的团删除后 $f(A)$ 的值是否变小，如果变小则删去。

单次执行上述操作时间复杂度为 $O(n \sum_{C \in A} |C|^2)$ 。在对每个点运行一段时间后，保证最后得到的状态为一个“满”的状态时输出，可以得到约 80 分。

2.5 算法 5

我们始终维护一个“满”的状态 A ，并希望 $|A|$ 尽量小。 A 初始即为 m 个团的集合，其中每个集合恰好包含一条边。反复执行以下操作：

1. 对于每个 A 中的团 C ，依次按某种随机顺序考虑所有不在 C 中的点 x ，查看当前 C 中的点是否与 x 均有边，如果有，那么将 x 加入 C 。显然这一步进行完后 A 仍然是一个“满”的状态。
2. 依次考虑所有 A 中的团 C ，查看如果把这个团从 A 中删去， A 是否仍然是一个“满”的状态，如果是，则把 C 从 A 中删去。显然操作后 A 仍为“满”的状态。
3. 依次考虑所有 A 中的团 C ，并以某种顺序遍历所有 C 中的点 x ，查看把 x 从 C 中删去后， A 是否仍然为一个“满”的状态，如果是，则把 x 从 C 中删去。显然操作后 A 仍为“满”的状态。

单次执行时间复杂度为 $O(n \sum_{C \in A} |C| + \sum_{C \in A} |C|^2)$ 。注意到在第 1 步和第 3 步中， $|A|$ 均不会变化；在第 2 步中， $|A|$ 可能会减少。所以在运行过程中 $|A|$ 一定单调不升，并且如果 A 不变， A 的形态会发生显著变化。所以，这是一个典型的调整算法 [1]。用这个算法对所有点运行一段时间后，可以得到 100 分。

2.6 注

一些验题人在第 4 和第 5 个点中给出了比标算（算法 5）更优的方案，但他们也都使用了调整算法。同时，在后 5 个点上，笔者并未使用足够长的时间运行出对于这个算法的最优解，所以在这个算法下，团的个数可以进一步降低。有兴趣的读者可以对这个问题进行更深入的研究。

3 参考文献

- [1] 邓明扬，《一类调整算法在信息学竞赛中的应用》，IOI2021 中国国家候选队论文