# 《这里是，终末停滞委员会。》解题报告

乐山高新区嘉祥外国语学校 黎莫轩

September 11, 2025

## Contents

# 1 题目

## 1.1 题目描述

A、B、C 三所学园共有 $n$ 名学生。有两种任务，分别有 $m_1, m_2$ 个，对于每个任务，都指定了两名参与者 $a_i, b_i$。

- 对于所有任务：当两名参与者都来自学园 C 时，获得 2 点收益。
- 仅对于第二种任务：如果两名参与者有恰好一名来自学园 C，则遭受 1 点损失。若两名参与者来自同样的学园，则遭受 2 点损失。

现在，某些学生入学的学园已经确定，你需要确定剩余的所有学生入学的学园，来最大化所有任务的总净收益（即，总收益减去总损失）。注意收益和损失可以叠加，例如如果完成第二种任务的两名学生都来自学园 C，则获得的净收益是 $2-2=0$。

然而，这个问题实在有些太简单了，因此我们给出正整数 $k$，接下来，把学生所属的学园对应的字母依次连接，得到一个长度为 $n$ 的仅包含 A、B、C 三种字母的字符串，你需要给出所有最大化净收益的方案中，字符串的字典序前 $k$ 小的方案。若方案数小于 $k$，则你需要给出所有方案。

## 1.2 数据范围

记 $m = m_1 + m_2$。

$2 \le n \le 10^5$，$0 \le m \le 10^5$，$\max(n, m) \times k \le 10^7$，$1 \le a_i < b_i \le n$。

- Subtask 1（2 分）：$m = 0$；
- Subtask 2（7 分）：$n \le 15$；
- Subtask 3（8 分）：$n \le 2 \times 10^4$，所有的任务都是第一种任务；
- Subtask 4（9 分）：$n \le 2 \times 10^4$，初始时，没有任何学生入学的学园已经确定；
- Subtask 5（24 分）：$n \le 2 \times 10^4$，所有的任务都是第二种任务；
- Subtask 6（20 分）：$n \le 2 \times 10^4$，$k = 1$；
- Subtask 7（18 分）：$n \le 2 \times 10^4$；
- Subtask 8（12 分）：无特殊限制。

本题有 Special Judge，如果输出正确的最优解和错误的方案，也可以获得对应子任务 50% 的分数。

# 2 解题过程

## 2.1 算法 1

对于 $m = 0$，显然最优解为 0，且任何方案都是最优解。DFS 输出字典序前 $k$ 小的方案即可。

可以通过 Subtask 1。

## 2.2 算法 2

对于 $n \le 15$，本质不同的任务只有 $O(n^2)$ 种，可以搜索所有可能的方案，并 $O(n^2)$ 计算其净收益。复杂度 $O(n^2 3^n)$。

可以通过 Subtask 2。

## 2.3 算法 3

考虑只有第一种任务的情况，则其中一种最优解显然是把所有未设定的人设置成 C。在这种最优解中，所有没有贡献的任务可以直接删掉。对于剩下的任务，其涉及的所有人在所有最优解中都必须是 C。在确定完这些人之后，剩下的人的选择是任意的。

可以通过 Subtask 3。

## 2.4 算法 4

考虑没有任何人的学园被确定的情况。显然其中一种最优解为把所有人设定成 C。对于第一种任务，要求两个人必须是 C；对于第二种任务，要求两个人分别是 CC 或 AB。

把任务看作无向边，得到一张无向图。考虑其中的一个连通块，若其包含第一种任务，则必须全部设为 C；否则，可以全部设为 C，若该连通块为二分图，则还可以将其 AB 相间染色。

可以通过 Subtask 4。

## 2.5 算法 5

前四个 Subtask 与本题正解的关系较小，因此给出的部分分分值较少。

接下来，我们考虑解决 Subtask 5 中无需构造方案的部分分。

第二类任务的贡献较为复杂，先前直接讨论的做法似乎不具有扩展性，我们考虑使用最小割解决这一问题。

我们知道，最小割存在一种等价定义：即，把所有点分入两个点集 $S, T$，其中源点必须分入 $S$，汇点必须分入 $T$，然后最小化所有从 $S$ 连向 $T$ 的边权总和。

然而，本题中我们需要把点分入 $A,B,C$ 三个集合，直接对其套用最小割的这个定义是无法做到的。因此，一个想法是：考虑拆点。我们对所有点 $u$ 拆出两个点 $x_u, y_u$。其中 $x_u$ 划入 $T$ 代表分入 $A$ 集合，$y_u$ 划入 $T$ 代表分入 $B$ 集合，$x_u, y_u$ 均划入 $S$ 代表分入 $C$ 集合。我们稍后会处理 $x_u, y_u$ 均被划入 $T$ 的情况。

注意到贡献的对称性，我们可以得到，对于一个合法的建边方案，如果存在边 $(p,q,w)$，则一定有对应的 $(q,p,w)$。因此，对于一个任务设计的点对 $a,b$，它们之间共有四条可以决策的边：$(x_a,x_b),(x_a,y_b),(y_a,x_b),(y_a,y_b)$。

考虑把这四条边的边权作为未知数解方程，不难得到：$(x_a,y_b,1),(y_a,x_b,1)$ 是合法的建边方案。注意这两条边都需要建立反边。

接下来考虑 $x_u, y_u$ 均被划分入 $T$ 的情况：发现在上面的建模中，$x_u, y_u$ 均被划分入 $T$ 不优于均被划分入 $S$。因此无需特殊处理这种情况。

最后，由于有些点的颜色已经固定了，我们通过添加 $S \to u$ 或 $u \to T$ 的无穷边强制其被划分入特定集合。

于是，我们得到了一个网络流建模，我们使用 Dinic 算法求解它。接下来，我们需要证明这一算法的时间复杂度。

**定理 2.5.1.** *对于一个流网络，设每个点的入边总容量为 $d_{in}(u)$，出边总容量为 $d_{out}(u)$。记 $S = \sum\limits_{u \in V} \min(d_{in}(u), d_{out}(u))$，则在其上增广的轮数是 $O(\sqrt{S})$ 的。*

*证明.* 首先，增广轮数达到 $\sqrt{S}$ 后，残量网络上的最短路长度不少于 $\sqrt{S}$。

接下来，若我们流 $d$ 条增广路，则可以在残量网络上找到 $d$ 条路径，满足不同时经过一条边和它的反边。对于残量网络上的一个非源汇点 $u$，它被经过的次数上界是 $\min(d_{in}(u), d_{out}(u))$（需要注意到的是：一次增广不会改变除源汇点以外的点的 $d_{in}$ 和 $d_{out}$）。于是选出的 $d$ 条路径的最大总长度与 $S+d$ 同阶，因此其中的最短一条的长度是 $O(\frac{S}{d})$ 量级的。

而最短路长度不少于 $\sqrt{S}$，因此 $d$ 最多是 $O(\sqrt{S})$ 量级，增广轮数也就是 $O(\sqrt{S})$ 量级。 $\square$

对于本题，我们立刻得到增广轮数为 $O(\sqrt{m})$。而所有非无穷边在单轮增广只会被访问一次，$S \to u$ 的边在单轮增广也只被访问一次，$u \to T$ 的边被访问的次数不超过最大流大小。因此复杂度为 $O(m\sqrt{m})$。

可以通过 Subtask 5 的 50% 部分分。

## 2.6 算法 6

沿用算法 5 的做法，尝试处理第一种任务。

由于最小割无法直接描述奖励，而只能描述惩罚，我们把答案直接增加 2，然后规定除了 CC 以外的组合会造成 2 的惩罚。

类似上一算法，设出 $(x_a,x_b),(x_a,y_b),(y_a,x_b),(y_a,y_b)$ 四条边的边权，并解方程，可以得到把四条边（及反向边）边权均设为 1 是合法的。

类似上一算法地讨论，可以发现 $x_u, y_u$ 均被划分入 $T$ 仍然不优于均被划分入 $S$。

同样，算法复杂度为 $O(m\sqrt{m})$。

可以通过全部 Subtask 的 50% 部分分。

## 2.7 算法 7

考虑如何构造字典序最小的方案。为了完成这一目标，我们需要先刻画合法的最小割的形态。

在此之前，让我们重新复习最大流最小割定理的证明：

**定理 2.7.1.** *最大流等于最小割。*

*证明.* 首先，我们证明任意割大于等于任意流。

令 $c(u,v)$ 为边的容量，$f(u,v)$ 为边的流量。

考虑一组割边 $C$，则割的权值为：

$$\begin{aligned}\sum_{(u,v)\in C} c(u,v) &\geq \sum_{(u,v)\in C} f(u,v) \\ &\geq \sum_{(u,v)\in C} f(u,v) - \sum_{(u,v)\in E\wedge u\in T\wedge v\in S} f(u,v)\end{aligned}$$

其中最后一行的式子等于 $S$ 集合到 $T$ 集合的流量，也就等于源点到汇点的流量。

接下来我们证明，对于最大流的方案，存在一组割的方案使得两个 $\geq$ 都取等。

也就是说：我们需要让 $S$ 集合到 $T$ 集合的所有边都为满流边，且不存在 $T$ 到 $S$ 的流量。

考虑把残量网络上源点能到的所有点加入 $S$ 集合，剩下的点加入 $T$ 集合。由于当前方案为最大流，由增广路定理，源点无法到达汇点，所以汇点一定在 $T$ 集合内，符合要求。

然后我们发现，在这组构造中，$S$ 集合到 $T$ 集合的所有边都为满流边，且不存在 $T$ 到 $S$ 的流量，否则源点能到达 $S$ 集合外的点，矛盾。 □

上述证明直接给出了一种求解最小割方案的算法。然而我们要求解最小字典序的方案，因此，需要发掘更多的性质。

**定理 2.7.2.** *对于最大流的一组方案，若一条残量网络中的边位于任何一个环中，则它对应的正边不属于最小割。*

*证明.* 考察我们刚刚给出的条件：割中的所有边都为满流边。若一条边不满流，则它一定不是割边。

若它满流，而它的反边处于环中，则对这个环做增广就会使得这条边不满流。因此它也一定不是割边。 □

这说明我们可以直接对残量网络执行缩点，在一个 SCC 内部的边一定不会成为最小割中的割边。此时，剩下的边要么是满流的，要么是没有流量的。于是，此时一个 SCC 一定全被划入 $S$ 或全被划入 $T$。

**定理 2.7.3.** *对于缩点后的 DAG，一个划分集合的方案合法，当且仅当不存在 $T$ 到 $S$ 的满流边或 $S$ 到 $T$ 的空边。*

*证明.* 我们已经说明了，最小割合法等价于 $S$ 集合到 $T$ 集合的所有边都为满流边，且不存在 $T$ 到 $S$ 的流量。容易发现上面的定理是这句话的等价表述。 □

我们建立一个 DAG，保留所有空边的反边，以及满流边，则合法的割满足不存在被染成 $T$ 的点能到达被染成 $S$ 的点。这已经是足够简洁的条件，接下来我们可以开始求解最小字典序的解了。

依照求解字典序问题的经典做法，我们每次决策尚能决策的编号最小的点的颜色。在决策过程中，我们维护出所有已经确定集合的点的所属集合。

首先尝试将其设为 A：这需要把 $x_u$ 染成 $T$ 并把 $y_u$ 染成 $S$。那么，如果 $x_u$ 已经被染成 $S$，或 $y_u$ 已经被染成 $T$，或者 $x_u$ 能到达 $y_u$，那么这就是不能做到的。否则，只需从 $x_u$ 出发 DFS 所有能到达的点染成 $T$，从 $y_u$ 出发在反图上 DFS 所有能到达的点染成 $S$ 即可。

设为 B、C 的讨论与上类似，在此不多赘述。

我们发现上面需要求解 $x_u$ 能否到达 $y_u$，这是 DAG 上的可达性问题，可以使用 bitset 在 $O(\frac{nm}{w})$ 复杂度内解决。然而开 $O(n)$ 个长度 $O(n)$ 的 bitset 的空间复杂度过大。由于询问在一开始已经确定，我们把所有询问进行分组，对每组运行 bitset 求可达性的算法。这样就在时间复杂度不变的情况下把空间复杂度降低到了线性。（实际运行中，分组的阈值可能设得较大，以减小运行时间的常数因子）

最终的复杂度为 $O(m\sqrt{m} + \frac{nm}{w})$。

可以通过 Subtask 6。

### 2.8 算法 8

我们需要求解字典序前 $k$ 小的解。实际上，上述按位确定答案的做法仍然可用，我们只需把贪心换成一个按照 A、B、C 顺序的 DFS 即可。求解单个方案的复杂度仍然是 $O(n+m)$ 的。

算法复杂度为 $O(m\sqrt{m} + \frac{nm}{w} + (n+m)k)$。

可以通过所有 Subtask。如果实现较劣（例如，没有对 DAG 可达性的询问离线处理），可以通过 Subtask 1 到 7。

## 3 命题思路

网络流是 OI 中的一类经典问题。在网络流问题中，最小割问题也占据了重要的地位。

在最小割问题中，“枚举所有边并解方程确定边权”，以及“对最小割形态的刻画”，都是有用的技巧，但在 OI 题目中较少被使用到。因此，笔者以这两种思想为核心命制了此题。希望做到本题的选手们能有所收获。

笔者对本题的解法在最后使用到了 bitset，然而此处的 bitset 只承担了求解可达性的工作。笔者认为，可能存在更优秀的做法以避免 bitset 的出现，做到更低的复杂度。但笔者未能得到好的方法，欢迎想到做法的同学和笔者联系。

# 4 参考资料


- Dinic 的几种复杂度：
  https://www.cnblogs.com/myee/p/dinic-algorithm.html；
- AHOI2009 最小割：
  https://www.luogu.com.cn/problem/P4126。



感谢**赵海鲲**、**叶隽霖**、**陈翔乐**同学参与了本题的验题工作。

感谢**叶开**学长与我关于本题的复杂度证明进行了交流。