《coneyisland》解题报告

湖南师范大学附属中学 颜桉

2023 年 11 月 25 日

目录

1 题目

1.1 题目大意

对于一张图 G 定义如下游戏：

游戏在足够聪明的两人 Alice 和 Bob 之间进行。有一枚棋子，Alice 先在 G 上任意选择一个节点，并将棋子放在该节点上。然后 Bob 和 Alice 轮流把棋子沿着 G 中的边移动到任意一个相邻的并且棋子之前没有停留过的节点上。Bob 先手，不能移动者输。

对于一张 n 个点的无向图 G 和正整数 k，图 G^k 定义如下：

- G^k 包含 nk 个点，分别为 $(1,1),(1,2),\cdots,(1,n),(2,1),\cdots,(2,n),\cdots,(k,1),\cdots,(k,n)$。
- 对于 $\forall i\in[1,k]$，$(i,u),(i,v)$ 之间有一条无向边，当且仅当 G 中 u,v 之间有一条无向边。
- 对于 $\forall i\in[1,k-1],u\in[1,n]$，$(i,u),(i+1,u)$ 之间有一条无向边。

有一个 n 个点的森林，初始包含 m 条边。有 q 次操作，操作分为三种：

- 1 u v：在 u,v 之间连一条无向边。保证连完后依然是一个森林。
- 2 u v：删除 u,v 之间的无向边。保证操作时 (u,v) 之间有一条无向边，
- 3 u k：记 T 为森林中 u 所在的树，求在 T^k 上进行上述游戏的胜者。

1.2 输入格式

第一行三个非负整数 n,m,q。
接下来 m 行，每行两个正整数 u,v，表示初始森林中 (u,v) 之间有一条无向边。
接下来 q 行，每行包含三个正整数，表示一次操作，格式如「题目描述」中所示。

1.3 输出格式

若干行，每行为 `Alice` 或 `Bob`，表示一次询问的胜者。

1.4 数据范围

本题使用捆绑测试，共有 10 个 Subtask：

- Subtask1（4 分）：$n\le 100,q\le 2000$，数据类型为 A1。
- Subtask2（13 分）：$n\le 200,q\le 2000$，数据类型为 A3。
- Subtask3（10 分）：$n\le 2000,q\le 2000$，数据类型为 C2
- Subtask4（11 分）：$n\le 5\times 10^4,q\le 2\times 10^4$，数据类型为 C1。
- Subtask5（13 分）：$n\le 5\times 10^4,q\le 2\times 10^4$，数据类型为 C2。
- Subtask6（15 分）：$n\le 10^5,q\le 10^5$，数据类型为 A3。
- Subtask7（4 分）：$n\le 10^5,q\le 10^5$，数据类型为 B1。

- Subtask8（7 分）：$n \le 10^5, q \le 10^5$，数据类型为 B2。
- Subtask9（12 分）：$n \le 2 \times 10^5, q \le 2 \times 10^5$，数据类型为 B3。
- Subtask10（11 分）：$n \le 2 \times 10^5, q \le 2 \times 10^5$，数据类型为 C3。

其中，数据类型包含两个参数，第一个为 A、B、C 之一，具体限制如下：

- A：保证只存在第三种操作。
- B：保证不存在第二种操作。
- C：无特殊限制。

第二个为 1、2、3 之一，具体限制如下：

- 1：保证 $k = 1$。
- 2：保证所有第三种操作中的 k 相同。
- 3：无特殊限制。

对于所有数据，保证 $1 \le n, q \le 2 \times 10^5, 0 \le m < n, 1 \le u, v \le n, 1 \le k \le 10^9$。

1.5 时空限制

时间限制 2s，空间限制 512MB。

2 解题过程

2.1 算法一

题目中所定义的游戏十分经典。我们有

引理 1. *Alice 必败当且仅当 G 存在完美匹配。*

证明. 一方面，假设 G 存在完美匹配。任取一组完美匹配，则无论 Alice 把棋子放在哪个节点上，Bob 都可以沿着匹配边把棋子移到与这个节点匹配的节点。接下来，因为 Alice 不能往回移，她只能把棋子移到一对新的匹配的点中的一个，然后 Bob 可以继续沿着匹配边移。这个过程会一直进行下去，直到 Alice 无法移动棋子。所以 G 存在完美匹配时 Alice 必败。

另一方面，假设 G 不存在完美匹配。任取一组最大匹配，假设 Alice 把棋子放在了任意一个不在这组最大匹配中的节点上。可以证明，每次 Bob 移动之后棋子都位于这组匹配中，否则 Bob 移完后棋子移动的路径构成了一条增广路，与最大匹配的条件矛盾。所以 Alice 每次都可以沿着匹配边移。这个过程会一直进行下去，直到 Bob 无法移动棋子。所以 G 不存在完美匹配时 Alice 必胜。 □

对于前两种操作，暴力进行加边和删边的操作；对于第三种操作，暴力把 T^k 建出来，使用带花树等算法判断其有无完美匹配。

时间复杂度 $\mathcal{O}(q(nk)^3)$，期望得分 4。

2.2 算法二

我们进行一些基本的观察。

首先，注意到 T^k 是一张二分图，因此可以用网络流代替带花树等算法，判断是否存在完美匹配的时间复杂度从 $\mathcal{O}((nk)^3)$ 降为了 $\mathcal{O}((nk)^{1.5})$。同时，我们考虑将 T 进行黑白染色。若黑点和白点的数量不相等，不难发现 T^k 中左部点和右部点的数量也不相等，不可能存在完美匹配，Alice 必胜。

另一个显然的观察是：当 $2 \mid k$ 时，显然 T^k 存在完美匹配，此时直接输出 `Bob` 即可。当 $2 \nmid k$ 时，对于 $\forall 2 \nmid k_0$，若 T^{k_0} 存在完美匹配，显然对于 $\forall 2 \nmid k, k \geq k_0$，$T^k$ 都存在完美匹配。所以，我们只需要找到最小的 k_0，使得 T^{k_0} 存在完美匹配即可。经过一些模拟，我们发现对于任意 T，$T^{|T|+1}$ 均存在完美匹配。猜测 $T^{|T|+1}$ 总存在完美匹配（接下来的解题过程可以推出这个结论），使用二分法求解最小的 k_0。

时间复杂度 $\mathcal{O}(n^3 \log n)$，期望得分 17。

2.3 算法三

考虑到经典的

引理 2 (Hall 定理)**.** *对于一张二分图 $G = \{V, E\}$，设其左部点集合和右部点集合分别为 X 和 Y。对于 $S \subseteq X$，设*

$$N(S) = \{v \mid \exists u \in X, such\ that\ (u, v) \in E\}$$

则 G 存在完美匹配的充要条件是对于 $\forall S \subseteq X$，有

$$|S| \leq |N(S)|$$

证明. 该引理的证明较为经典，读者可以自行搜索。 □

我们转而考虑能否找到一个 S，使得 $|S| > |N(S)|$。首先，我们对需要讨论的情况进行一些缩减。如算法二中所提及的，我们注意到两个事实：

1. 当 $2 \mid k$ 时，T^k 存在完美匹配，Bob 必胜。
2. 当 $2 \nmid k$ 时，将 T 进行黑白染色。若黑点和白点的数量不相等，则 Alice 必胜。

因此，我们只需要讨论 $2 \nmid k$ 且 T 黑白染色后黑点和白点数量相等的情况。对于 $\forall i \in [1, k]$，我们把节点 $(i, 1), (i, 2), \cdots, (i, n)$ 称作**第 i 层**的节点。设第 i 层中我们选择的节点在 T 中对应节点的集合为 S_i。我们以最大化 $|S| - |N(S)|$ 为目标，尝试寻找 S_i 的一些性质。我们的方式是通过调整法来说明存在至少一个最优解，满足某些特定的条件。因此，为了保持证明过程的简洁性，下文中的「最优解满足」均代表「存在至少一个最优解满足」。

在接下来的推导中，一个需要注意的地方是：因为 S_i 是定义在 T 上的，所以 $N(S_i)$ 以及下文中会出现的 $R(S_i)$ 也是定义在 T 上的；与之相对地，S 和 $N(S)$ 则是定义在 T^k 上的。

为了方便接下来的推导，不妨设 $S_0 = S_{k+1} = \varnothing$。

我们发现，每一层的 S_i 的选择仅会影响相邻的两层，范围较小。于是我们可以大胆猜测：最优解中奇数层和偶数层的结构分别相同。形式化地说，即

引理 3. 对于 $\forall i \in [2, k-1]$，有 $S_{i-1} = S_{i+1}$。

当然，这并不是一件容易证明的事情。我们可以先从 $k = 3$ 的情况着手，考虑证明

引理 4. $k = 3$ 时，最优解满足 $S_1 = S_3$。

证明. 考虑调整法。我们将证明：令 $S_1 \leftarrow S_3$，或者令 $S_3 \leftarrow S_1$，其中总有一个是不劣的。对于初始情况，我们有

$$M = |S| - |N(S)| = |S_1| + |S_2| + |S_3| - |N(S_1) \cup S_2| - |S_1 \cup N(S_2) \cup S_3| - |S_2 \cup N(S_3)|$$

而对于令 $S_1 \leftarrow S_3$ 后的情况，我们有

$$M_1 = 2|S_1| + |S_2| - 2|N(S_1) \cup S_2| - |S_1 \cup N(S_2)|$$

同理

$$M_3 = |S_2| + 2|S_3| - 2|S_2 \cup N(S_3)| - |N(S_2) \cup S_3|$$

取平均数，得

$$\frac{M_1 + M_3}{2} = |S_1| + |S_2| + |S_3| - |N(S_1) \cup S_2| - \frac{|S_1 \cup N(S_2)| + |N(S_2) \cup S_3|}{2} - |S_2 \cup N(S_3)|$$

显然，$\frac{M_1+M_3}{2} \geq M$。这意味着两个调整中总有一个是不劣的。 □

当 $k > 3$ 时，我们考虑以奇数层为主进行构造。当确定完奇数层之后，偶数层实际上是独立的。所以若能证明所有奇数层相同的话，所有偶数层相同就是水到渠成的事了。对于 $k = 5$，若仿照引理 4 中的的方法，我们可以得到

引理 5. $k = 5$ 时，以下四个调整中中总有一个是不劣的：

- 令 $S_1 \leftarrow S_3$；
- 令 $S_3 \leftarrow S_1$；
- 令 $S_3 \leftarrow S_5$；
- 令 $S_5 \leftarrow S_3$。

证明. 该引理的证明方法与引理 4 并无二致，且过程较为繁琐，在此略去。 □

可是，只有引理 5 并不足以说明最优解满足 $S_1 = S_3 = S_5$，原因是可能出现 $S_1 = S_3 \neq S_5$ 或 $S_1 \neq S_3 = S_5$ 的情况。为了修补这个漏洞，我们需要

引理 6. 对于某个解，若存在一个 $1 \leq p < k, 2 \nmid p$，使得 $S_1 = S_3 = \cdots = S_p \neq S_{p+2} = \cdots = S_k$，那么总存在一个不劣的调整，使得调整完后所有奇数层的 S_i 相同。

证明. 设 $f(P, Q)$ 表示在 $k = 3$ 的情况中钦定 $S_1 = P, S_3 = Q$ 的情况下 $|S| - |N(S)|$ 的最大值，$g(P, Q)$ 使得 $f(P, Q)$ 取到最大值的任意一个 S_2。不妨设 $f(S_1, S_1) - |N(S_1) \cup g(S_1, S_1)| \geq f(S_k, S_k) - |N(S_k) \cup$

$g(S_k, S_k)|$。我们发现每次可以把 p 右侧部分的某个 $S_{q-1} = S_k, S_q = g(S_k, S_k), S_{q+1} = S_k$ 中右边的两层拿出来，变为 $g(S_1, S_1), S_1$，然后插进 p 左侧部分的某个位置中，这样做显然不劣。于是我们可以把解的形式变为 $S_1 = S_3 = \cdots = S_{k-2} \neq S_k$。

固定前 $k-2$ 层，考虑如何去第 k 层最优。若存在 $S' \neq S_1$ 使得 $f(S_1, S') > f(S_1, S_1)$，那么可以用 S' 把 $S_3, \cdots, S_{k-2}$ 全部换掉，然后继续进行类似的过程。

显然，这个过程是有限的。我们最终肯定能够找到一个 S_0，使得 $\max f(S_0, *) = f(S_0, S_0)$。那么此时整个调整结束，最终 $S_1 = S_3 = \cdots = S_k = S_0$。 □

有了这些铺垫，我们终于可以尝试证明引理 3 了。

证明. 考虑任意的 S_i 相同的连续段。若存在连续两个段的长度 > 1，则可以使用引理 6 中的方法调整。假设一个长度 > 1 的段的右端点为 p，下一个长度 > 1 的段的左端点为 q。那么 $p - q \geq 4$，并且 p, q 之前肯定存在一个形如「谷底」的位置 r，满足把 $[r-2, r+2]$ 这五层截取出来当作 $k = 5$ 的情况时，引理 5 中的第一种和第四种调整都会变劣的，于是只能调整 S_r，新增一个长度 > 1 的连续段。结合引理 3 中我们对两侧的调整，最终我们可以使得解变为形如引理 6 中的形式，于是可以调整得到一组奇数层全部相同的解，进而可以得到一组满足对于 $\forall i \in [2, k-1]$，有 $S_{i-1} = S_{i+1}$ 的解。 □

于是引理 3 得证。这意味着我们只需要考虑奇数层所选择的节点 S_{odd} 和偶数层所选择的节点 S_{even}。此时我们对于 S_i 的刻画已经较为清晰了，由于 k 已经在询问中给定，我们可以使用一个简单的树形 dp 来判断是否存在一组 $(S_{\text{odd}}, S_{\text{even}})$，使得最终的 $|S| - |N(S)| > 0$。我们再使用 set 来维护森林中的边，就可以做到 $\mathcal{O}(\log n)$ 完成前两种操作。

时间复杂度 $\mathcal{O}(qn \log n)$，期望得分 42。

2.4 算法四

对于 $k = 1$ 和 k 固定的情况，我们都可以使用动态 dp 和 Link-Cut-Tree 来维护上文中的 dp，区别是状态设计和转移的复杂程度不同。因为扩展性不佳，并且与本题正解没有太大关联，我们省去关于该算法细节的推导。

时间复杂度 $\mathcal{O}(n \log^2 n)$ 或 $\mathcal{O}(n \log n)$，期望得分 49，结合算法三期望得分 62。

2.5 算法五

在 k 不固定的情况下，如果想要避免每次都进行 $\mathcal{O}(n)$ 的 dp，我们就必须弱化 k 在一次询问中的影响，于是我们把问题转化为找到最小的奇数 k，使得 T^k 存在完美匹配。为了方便，设 $k = 2m + 1$。

考虑继续发掘性质。一个直观的感受是，如果一个点周围的点都被覆盖了，则再选上这个点显然不劣。形式化地说，对于 T 中节点的一个子集 P，记 $R(P) = \{u \mid N(\{u\}) \subseteq P\}$，有

引理 7. $R(S_{even}) \subseteq S_{odd}$。

证明. 假设存在 $u \in R(S_{\text{even}})$ 使得 $u \notin S_{\text{odd}}$。那么 $N(\{u\}) \subseteq S_{\text{odd}}$，所以若令 $S_{\text{odd}} \leftarrow S_{\text{odd}} \cup \{u\}$，显然总的 $|S| - |N(S)|$ 会增加 m，所以选上 u 不劣。 □

进一步地，我们有

引理 8. $S_{even} = R(S_{odd})$。

证明. 假设存在 $u \in S_{\text{even}}$，使得 $u \notin R(S_{\text{odd}})$。设 $c = |S_{\text{odd}} \cup N(u)| - |S_{\text{odd}}|$，则 $c > 0$。在 S_{even} 中删去 u，$|S| - |N(S)|$ 的增量为 $-m + (m+1)c > 0$。所以删去 u 总是优的。 □

这两个引理使得我们只需要关心一个集合，即 S_{odd}，不妨记其为 U。我们有

$$\begin{aligned}|S| - |N(S)| =&(m+1)|U| + m|R(U)| - (m+1)|N(U)| - m|U| \\ =&|U| - (m+1)|N(U)| + m|R(U)|\end{aligned}$$

那么，算法三中的 dp 需要改为求最小的 m，使得对于任意 U，上式的值恒非正。因为 T 中黑白点的数量相等，所以 $|N(U)| = |R(U)|$ 的情况是没有意义的，所以我们可以把问题转化为最大化

$$\frac{|U| - |N(U)|}{|N(U)| - |R(U)|}$$

很遗憾，对于这个式子，笔者并没有想出什么高效的解决办法。是否还有值得发掘的性质呢？注意到 $U \cup R(U)$ 在树上形成了若干个连通块，我们对这些连通块的性质进行研究，可以得到

引理 9. 不妨设 U 中的点为黑色节点，那么 U 中的点构成了一个黑色节点的连通块。即，那么不存在一条链上依次排列的三个黑色节点 u, v, w，使得 $u, w \in U \wedge v \notin U$。

证明. 假设存在这样的一个点，那么考虑以 v 为根，把 U 中所有点按照其在 v 的哪个儿子的子树中分为若干个集合。此时，任意选取 v 的两个不同儿子的子树，其中若都包含 U 中节点，记为 U_{v_1} 和 U_{v_2}，那么 $N(U_{v_1})$ 和 $N(U_{v_2})$ 的交集必然为空。于是 $\frac{|U|-|N(U)|}{|N(U)|-|R(U)|}$ 可以看作所有 $\frac{|U_v|-|N(U_v)|}{|N(U_v)|-|R(U_v)|}$ 的平均值。显然，将 U 替换为其中最大值对应的 U_v 必然不劣。 □

继续进行研究，自然地，我们可以得到

引理 10. $|N(U)| - |R(U)| = 1$。

证明. 不妨设 U 中的点为黑色节点。假设 $|N(U)| - |R(U)| = k > 1$。对于每个不在 $U \cup N(U)$ 中的连通块，由引理 9 知其必然恰好包含一个 $N(U) \setminus R(U)$ 中的点。记这个连通块中黑色节点的个数减去白色节点的个数的值为 s，那么如果将这个连通块中的所有黑点都添加到 U 中，$|U| - |N(U)|$ 会增加 $s+1$，而 $|N(U)| - |R(U)|$ 会从 k 变为 $k-1$。所以这样操作会变劣等价于

$$\frac{|U| - |N(U)|}{k} > \frac{|U| - |N(U)| + s + 1}{k-1}$$

即

$$|U| - |N(U)| + k(s+1) < 0$$

对于所有的连通块累加求和，得到

$$k(|U| - |N(U)|) + k\left(\sum(s+1)\right) < 0$$

又因为 T 中黑白节点个数相等，有

$$|U| - |N(U)| + \left(\sum (s+1)\right) = 0$$

矛盾。于是肯定存在一个连通块，使得将这个连通块中的所有黑点都添加到 U 中不劣。 □

有了引理 9，我们可以彻底改变思路！我们枚举这个这个唯一的点 r，考虑以 r 为根，那么我们选择的点肯定是 r 的某些子树中的全部节点。不妨设 r 为白色节点，最优的选择显然是所有删掉 r 后白色节点数量大于黑色节点数量的子树。所有的 r 对应的 $|U| - |N(U)|$ 的最大值就是最小的合法 m。注意到，所有子树中白色节点数量与黑色节点数量之差的和为 -1，经过简单的数值推导，我们发现最小的合法的 k 的值可以简单地表示为：删去 r 之后，所有子树中黑色节点与白色节点数量之差的绝对值之和。

这个信息的维护看起来比算法三中的 dp 方便得多，事实上的确如此。现在我们需要对一个黑白染色后的森林进行加边、删边操作，以及对于森林中的每棵树维护

$$\max_u \left\{ |\text{size}(u)| + \sum_{v \in \text{son}(u)} |\text{size}(v)| \right\}$$

其中我们任意指定一个节点为根，son(u) 表示 u 的儿子集合、size(u) 表示 u 子树中节点的权值和，其中黑点权值为 1、白点权值为 -1。

由于加边操作中可能会存在 u, v 已经被我们染为了相同颜色的情况，我们需要对于每个点分别维护其权值为 1 或 -1 上式的值。

考虑 Link-Cut-Tree。使用 LCT 维护子树信息的一个通用套路是：对于实儿子和虚儿子分开维护。注意到自身和实儿子 size 的绝对值处理与辅助树上的祖先节点有关，直接做不好使用 splay 维护。注意到 $|x| = \max\{x, -x\}$，所以 $|x| + |y|$ 可以看作 $\max\{x+y, x-y, -x+y, -x-y\}$。对于这四种情况分别记录最大值，就不需要担心祖先节点对 size 绝对值的影响了。所以对于每个节点 u ，我们需要记录如下信息：

- 其所有虚儿子 size 之和。
- 其所有虚儿子 size 绝对值之和。
- splay 树中，u 的子树对应的链上节点权值和虚儿子 size 之和。
- |size(u)| 和 |size(v)| 符号的四种取值下的最大答案，其中 v 为 u 的实儿子。
- 因为需要支持翻转，对于每条实链，即每棵 splay，我们需要维护其翻转之后的上述信息。
- 虚儿子的答案需要上传，所以我们还需要维护一个集合，表示其每个虚子树答案的最大值。这个集合需要支持加入和删除元素，以及查询最大值，可以对每个节点开一个 set 或者用两个优先队列模拟。

本题的代码细节繁多，在此不再赘述。选手在编写代码的时候需要仔细思考、多加注意。

LCT 本身的操作次数是 $\mathcal{O}(n \log n)$。因为我们对每个节点开了一个 set（或者两个优先队列），总的时间复杂度为 $\mathcal{O}(n \log^2 n)$，期望得分 100。

3 总结与展望

本题的 idea 和 solution 由笔者与笔者的学长陈永志（EmptySoulist）共同讨论得到。感谢他对这道题作出的贡献，以及他同意将这道题分享到集训队互测中。此外，笔者在造题过程中有过与集训队队员杨敏行、肖子尧的讨论，感谢杨敏行同学指出笔者证明中的错误。

本题从一个经典的博弈问题出发，转化为对于满足特殊性质的图，即 T^k，判定是否存在完美匹配的问题，又通过 Hall 定理转化为一个贪心问题。这个贪心问题的最优解具有许多优美的性质，选手对于性质的刻画越清晰，就能够拿到越高的分数。在一点点接近最优解的过程中，惊人地发现其具有如此简洁的形式，即可以表示为子树的带权 size 和，于是可以使用 Link-Cut-Tree 维护加边删边操作。同时，本题的代码难度也并不低，LCT 的若干细节需要选手仔细思考。总的来说，笔者认为，本题不失为一道好题。

笔者认为，本题的一大遗憾之处在于，如果不在一开始从 Hall 定理入手考虑，似乎很难获得较高的分数。除此之外，笔者也没有得到数据类型 B 的一些高效做法。希望在互测中实力强劲的集训队队员和精英培训选手能够弥补这两个空缺。

4 参考资料

[1] 维基百科，Hall's marriage theorem，https://en.wikipedia.org/wiki/Hall