HAMILG解题报告

宁波市镇海蛟川书院 施舟行

1 题目描述

1.1 题目来源

HAMILG from Codechef July Challenge 2015.

1.2 题目大意

两个玩家Askar和Bob，正在用一个在无向图G上的硬币进行游戏。这个游戏如下进行：

- Ascar选择一个起始点，并把硬币放在这个点上。
- 接下来，两个玩家轮流操作。Bob先进行操作。
- 轮到每个玩家操作时，他需要把硬币沿着一条边移至另一点。
- 硬币不能重复到达同一点。
- 无法操作的玩家将输掉这个游戏。

称一个点v为胜利点，当且仅当Askar能够通过选择v为起始点来获胜。假设两个玩家都按最优策略进行操作。给出有N个点，M条边的无向图G，求出有多少胜利点。

数据范围：$N \leq 2000, M \leq 10^6$。

2 算法讨论

设Askar选择的起始点u为胜利点，且接下来两名玩家各操作K次，Bob每次移动硬币后的终点为$x_1, x_2, \cdots, x_K$，Askar每次移动后的终点为$y_1, y_2, \cdots, y_K$。此时，x_i与$y_i(1 \le i \le K)$可以相互匹配，匹配数为K。若此时Bob还可以操作，设可将硬币移向点v，则点u可与x_1匹配，$x_i(2 \le i \le K)$可与y_{i-1}匹配，y_K与v匹配，此时匹配数为$K+1$，较之前获得的匹配更大。而若x_i与y_i的匹配，属于图G的某一个最大匹配，且点u是未被匹配的点，Bob就无法找到下一步操作的点v。于是可以发现，点u为胜利点的充要条件是：在任意一个图G的最大匹配中，点u为孤立点，也就是未被匹配。

我们可以使用带花树算法(*Edmonds' Blossom Algorithm*) 求出图G的一种最大匹配。以普通方式实现的带花树算法复杂度为$O(NM)$（实际测试时往往无法达到上界），经过优化的算法可将复杂度将至$O(N^{0.5}M)$。

2.1 基于多次匹配的判定算法

使用带花树算法求出图G的一种最大匹配后，我们可以枚举点i，并判定其是否是胜利点。若点i为胜利点，则在原图中删去点i后，图的最大匹配较原匹配减少，也就是点i一定会出现在所有的最大匹配中。这个算法的复杂度为$O(N^2M)$。

2.2 改进后的算法

使用带花树算法求图的最大匹配复杂度较高。实际上，并没有必要多次求最大匹配。我们可以在第一次匹配的基础上，枚举每一个孤立点，仿照带花树算法寻找匹配的过程，从这个孤立点出发寻找增广路。假设原先有匹配：

$$(x_1, y_1), (x_2, y_2), \cdots, (x_j, y_j)$$

现在从孤立点u出发，依次沿图G上的边走过$u, x_1, y_1, x_2, y_2, \cdots, y_i$，此时可以修改匹配为：

$$(u, x_1), (y_1, x_2), \cdots, (y_{i-1}, x_i), (x_{i+1}, y_{i+1}), \cdots, (x_j, y_j)$$

此时点y_i成为了孤立点，而匹配数没有发生改变。可以发现，每一条长度为偶数的增广路的终点，都可能在最大匹配中成为孤立点，也就是游戏中的胜利点。

除了在寻找增广路过程中所访问到的，满足其配偶先被访问的点是符合要求的胜利点，在带花树花*(blossom)*中的所有点，都可以通过调整增广路经过花上点的顺序，使它们都可能是偶数长增广路的终点。因此，在缩花的时候，把在花上的点都标记为胜利点即可。

算法复杂度与带花树算法复杂度相同，为$O(NM)$。