

# 密码锁解题报告

雅礼中学 毛啸

## 1 试题来源

Universal Online Judge Round(UR) #12 Problem B

<http://uoj.ac/problem/181>

## 2 题目描述

在你的帮助下，跳蚤国王发现把小队排列成理想情况是不可能的，于是他放弃了让小队重新排列的计划，直接下令强攻实验室。

在经过一番血战之后，跳蚤军队成功攻克了实验室中的大部分区域并把picks博士与他的猴子们逼入了最后一间房间中——同时也是存储了绝大部分资料和研究成果的最重要的房间。

这间房间的建造利用了跳蚤夸克神奇的力量，因此跳蚤国王发现强行闯入房间是不可能的，而唯一的入口——房间正门却被一个密码锁锁住了。

在这个密码锁上画了一张 $n$ 个点的完全图，其中每一条边旁边都写了一个数字。跳蚤国王发现只有 $m$ 条边对应的数字不是5000（其他数字都是5000）。

经过一番探索，跳蚤国王发现这个密码锁的密码等于这样一个问题的答案：

密码锁上给出了一张 $n$ 个点的完全图，现在要给这个完全图的每一条边随机定向成一个有向图。对于一条边 $(i, j) (i < j)$ ，这条边的方向是 $i$ 到 $j$ 的概率是 $\frac{\text{num}_{i,j}}{10000}$ ， $\text{num}_{i,j}$ 指这条边旁边的数字，否则就是 $j$ 到 $i$ 。在随机定向后，设这张有向图的强连通分量数目为 $x$ ，求 $x \times 10000^{n(n-1)}$ 的期望，可以证明该期望值一定是一个整数。因为答案可能很大，所以只要求出这个答案对 $9982443537 \times 17 \times 2^{23} + 1$ ，一个质数取模后的结果。

跳蚤国王发现这个问题并不是非常简单，于是他让你——这附近最著名的民间科学家来帮他计算这个问题的答案。

### 3 数据范围

$$n \leq 38, m \leq 19.$$

### 4 算法简介

这题是本人为UR出的一道题，考场上无人解出此题，从得分情况看属于UR B题中的较难题。

#### 4.1 算法一

枚举每条边的定向，然后暴力求出强连通分量数。

什么你说你不会求强连通分量数？直接用Floyd跑一遍求传递闭包，把所有的互相能到达的点都缩起来就可以了。当然你也可以BFS。

如果使用 $O(n + m)$ 的求强连通分量的算法，比如Tarjan算法，则时间复杂度为 $O(2^m(n + m))$ 。

#### 4.2 算法二

懂逆元的同学都应该知道模意义下统计一个东西的概率，我们可以先统计出强连通分量数的期望，最后乘上 $10000^{n(n-1)}$ 即可。如果不知道逆元可以参考维基百科。可以算出10000在模998244353下的逆元为796898467。

注意到定向后的图是个有向完全图，即竞赛图，竞赛图的强连通分量缩点之后，一定是一个链状的图，满足每个点和后面的点都有连边。

如何统计答案呢，设这张图为 $G(V, E)$ ，注意到答案即为链的长度，链的长度的统计可以转化为链的前缀的数量减一，这条链的前缀显然就是一个点集 $S \subset V$ ，满足所有的 $S$ 与 $V - S$ 之间的连边都是从 $S$ 中的点连出去的，我们定义这样的点集为割集（只是这篇题解中临时引入的定义），那么我们统计割集数量即可。

暴力枚举所有可能的点集 $S$ ，然后直接计算所有 $S$ 与 $V - S$ 之间连边为 $S$ 连出的概率，最后乘起来。

时间复杂度 $O(2^n m)$ 。

### 4.3 算法三

我们发现 $m$ 比较小，因此最好能想出一个在指数意义上和 $n$ 无关的算法。

对于一个点集，它是割集的概率无非就是这个点集中的点与点集外的所有点的所有连边都是从这个点集连出去的概率，也就是每条边是连出去的概率乘起来。对于概率非0.5的边我们称其为特殊边，对于大小为 $x$ 的点集，如果其中没有任何特殊边则是割集的概率为 $0.5^{x(n-x)}$ ，如果其中存在一条特殊边，设这条边从 $x$ 连出的概率为 $P$ ，那么这个概率要乘上 $2P$ ，也就是这个特殊边的贡献。我们不妨统计出每个大小的连通块，其中所有特殊边的贡献，最后对每个大小的连通块乘上 $0.5^{x(n-x)}$ 再加起来即可。

一个想法就是枚举有贡献的边的集合，如果一条边有贡献，那么它的两个端点必然有且仅有一个在 $S$ 中，那么我们可以考虑黑白染色，如果染色失败那么一定不存在这种情况，否则我们对于某个连通块，要么黑点在 $S$ 中，要么白点在 $S$ 中，并且每种情况的贡献都容易算出，所以我们通过一个 $O(n^2)$ 的类似背包问题的DP，即可算出每种大小的连通块的贡献总和。

时间复杂度为 $O(2^m n^2)$ 。

### 4.4 算法四

我们发现之前的复杂度在于背包DP上，对什么进行背包DP呢，对每个连通块进行背包DP，等等你说每个连通块？

一个连通块，如果有 $m$ 条边，那么最多有多少个点， $m+1$ 个！我们可以对每个连通块直接枚举哪些点在 $S$ 中，然后计算贡献，最后再来一次类似背包的DP。于是这题就被我们愉快的解决了。代码又短又优美，不比暴力要长。

时间复杂度为 $O(2^{m+1}m + n^2)$ 。