# Bingo! 解题报告

长沙市雅礼中学 刘研绎

## 1 试题来源

Codeforces MemSQL Start[c]UP 2.0 – problem D Bingo!

http://codeforces.com/contest/458/problem/D

## 2 试题大意

bingo是一种风靡世界的游戏。对原版游戏厌倦后你接触到了一个新版本的bingo。她在一个$n \times n$的棋盘上进行，每个格子中有一个范围在$1 \dots m$中的数，不会有两个格子拥有相同的数。现在你在$1 \dots m$中选取了$k$个数，对于每一个你选出的数，如果他在棋牌上出现了，就将其所在的格子涂黑。操作完后你将得到$2^{a+b}$的得分，其中$a$为完全被染黑的行数，$b$为完全被染黑的列数。

你很好奇，如果棋盘上的数随机选取，并且玩家的$k$个数也是随机选取，游戏的期望得分是多少。

输出答案和$10^{99}$中较小的那个。

相对误差小于$10^{-9}$即当做正确。

$n \le 300, n^2 \le m \le 10^5, n \le k \le m$

## 3 算法介绍

### 3.1 常规思路

首先按照常规思路思考。

题目描述中棋盘和我们需要选的$k$个数均是随机的，而实际上我们可以在随机之后将棋盘上的数重新编号，因为此时各个数之间没有本质区别，很容易

发现不论随机了怎样的棋盘，所得到的期望分数均不变。不妨设棋盘上填入了$1\ldots n^2$。

之后就是选数的问题了，不妨枚举$k$个数中，$1\ldots n^2$选中了s个，此时的概率是

$$\sum_{s=0}^{k}\frac{\binom{n^2}{s}\binom{m-n^2}{k-s}}{\binom{m}{k}} \tag{1}$$

即在$n^2$中取$s$个乘上在剩下的数中去$k-s$个得到在$1\ldots n^2$中选$s$个的总数，除以总的选$k$个数的方案数即得到概率。

继续采用常规思路，接下来我们应该统计给一个$n\times n$的网格随机染黑$s$个格子的期望得分，直接的思路毫无疑问是枚举完全染黑的行数$i$和列数$j$，再通过对答案的贡献乘以概率计算期望值，可以得到：（令$t$为完全染黑$i$行$j$列所花费的黑格子数）

$$\sum_{i}\sum_{j}^{t=n(i+j)-ij}\frac{2^{i+j}\binom{n}{i}\binom{n}{j}F(n-i,n-j,s-t)}{\binom{n^2}{s}} \tag{2}$$

即计算贡献为$2^{i+j}$的方案数，除以总方案数得到概率，乘上贡献得到期望。

其中函数$F(n-i,n-j,s-t)$表示在一个$(n-i)\times(n-j)$的棋盘上染黑$s-t$个格子，没有任意一行任意一列被完全染黑，有多少种方案。

综上，可以得到答案为：

$$(1)\times(2)=\sum_{s=0}^{k}\frac{\binom{n^2}{s}\binom{m-n^2}{k-s}}{\binom{m}{k}}\times\sum_{i}\sum_{j}^{t=n(i+j)-ij}\frac{2^{i+j}\binom{n}{i}\binom{n}{j}F(n-i,n-j,s-t)}{\binom{n^2}{s}}$$

但我们遇到了很多问题，单单是将所有情况枚举一遍就需要$O(mn^2)$的复杂度，已经超时。而且$F$函数的计算也是困难重重，几乎算不出来。

常规思路到此为止。

## 3.2 一个小技巧

考虑在计算函数$F$时，我们最大的问题就是如何保证没有任意一行任意一列被完全染黑。

如何避免这个问题?

普通的条件往往容易让人忽略，题目有一个特殊性质：如果有$i$行$j$列被完全染黑那么贡献$2^{i+j}$给答案，而不是形如$v(i+j)$ 然后将函数$v$输入给我们，这里有可能是一个突破口。

我们不保证没有任意一行一列完全被染黑的情况下计算$F$，会有合法的方案被多次统计，如果我们能够让每一个合法的方案统计次数和贡献有关联，或许我们可以通过方案被重复统计的次数直接算出权值，不需要去重。

可以发现，如果我们让剩下的黑格子随便放，不限制是否有被完全染黑的行和列，那么每种有i行j列被完全染黑的方案恰好被统计$2^{i+j}$次，也就是说我们不用乘上权值了。

那么(2)变为：

$$\sum_{i}\sum_{j}^{t=n(i+j)-ij}\frac{\binom{n}{i}\binom{n}{j}\binom{n^2-t}{s-t}}{\binom{n^2}{s}} \tag{3}$$

选出被染黑的$i$行$j$列的方案数乘以剩下随便放的方案数，得到每种方案算一次的方案数。

证明：对于一个恰好有$i$行$j$列被染黑的方案，对于任意一个行、列染黑的情况均为其子集的方案中会被计算一次，即总共计算$2^{i+j}$次，和贡献相等，因此(2)=(3)。

## 3.3 另一部转化

可以得到答案为：

$$\begin{aligned}
(1)\times(3) &= \sum_{s=0}^{k}\frac{\binom{n^2}{s}\binom{m-n^2}{k-s}}{\binom{m}{k}}\sum_{i}\sum_{j}^{t=n(i+j)-ij}\frac{\binom{n}{i}\binom{n}{j}\binom{n^2-t}{s-t}}{\binom{n^2}{s}} \\
&\Rightarrow \sum_{s=0}^{k}\frac{\binom{m-n^2}{k-s}}{\binom{m}{k}}\sum_{i}\sum_{j}^{t=n(i+j)-ij}\binom{n}{i}\binom{n}{j}\binom{n^2-t}{s-t} \\
&\Rightarrow \frac{1}{\binom{m}{k}}\sum_{i}\sum_{j}^{t=n(i+j)-ij}\binom{n}{i}\binom{n}{j}\sum_{s=t}^{k}\binom{m-n^2}{k-s}\binom{n^2-t}{s-t}
\end{aligned} \tag{4}$$

即使我们避免了函数$F$的计算，直接计算这个仍然会超时。我们需要更好的方法。

令$l = s - t$，则

$$\Rightarrow \frac{1}{\binom{m}{k}} \sum_{i} \sum_{j}^{t=n(i+j)-ij} \binom{n}{i}\binom{n}{j} \sum_{s=t}^{k} \binom{m-n^2}{k-t-l}\binom{n^2-t}{l}$$

可以发现，后面的式子相当于枚举在$n^2 - t$中选$l$个，剩下$k - t - l$在$m - n^2$中选，这相当于在$m - n^2 + n^2 - t$中选$k - t$个的总方案数。

$$\Rightarrow \frac{1}{\binom{m}{k}} \sum_{i} \sum_{j}^{t=n(i+j)-ij} \binom{n}{i}\binom{n}{j}\binom{m-t}{k-t}$$

这里就是我们的最终答案了。

## 3.4 总览

答案为：

$$\frac{1}{\binom{m}{k}} \sum_{i} \sum_{j}^{t=n(i+j)-ij} \binom{n}{i}\binom{n}{j}\binom{m-t}{k-t}$$

虽然答案可以达到$10^{99}$，但其实不需要写高精度，因为只判断相对误差所以浮点数类型就能胜任。如果担心中间结果太大可以取log进行计算。

可以预处理阶乘快速计算组合数。

至此问题便圆满解决。

复杂度：$O(n^2)$。