# Line 解题报告

成都七中 王迪

## Contents

# 1 题目大意

统计满足如下要求的 $N$ 行 $M$ 列 01 矩阵数目：

- 每一行都不能有连续 $P$ 个 0；
- 全是 0 的列数目不能超过 $Q$。

有 5% 的数据，$P=1$；
有 10% 的数据，$N\times M\le 20$；
有 15% 的数据，$N\le 2$，$M\le 10^6$；
有 10% 的数据，$N\le 2$；
有 20% 的数据，$N\le 4$，$P\le 2$，$Q\le 2$；
对于 100% 的数据，$N\le 8$，$M\le 10^{18}$，$P\le 3$，$Q\le 3$。

# 2 基础知识

这篇题解假定大家了解矩阵、矩阵乘法，以及如何用矩阵乘法优化递推。请不了解的同学自行学习。

# 3 算法分析

这是一个组合计数问题，入手点很多，我们根据数据规模来进行分析。

# 4 5% 的算法

注意到有 5% 的数据满足 $P=1$，即不能有连续 1 个 0，也就是说只能填 1，所以答案就是 1。

时间复杂度 $O(1)$，期望得分 5 分。

# 5 15% 的算法

有 10% 的数据满足 $N\times M\le 20$，这意味着我们可以枚举所有可能的矩阵进行判断。

时间复杂度 $O(2^{NM}NM)$，结合之前的算法可以得到 15 分。

# 6 30% 的算法

有 15% 的数据满足 $N\le 2$，$M\le 10^6$，我们可以考虑动态规划。为了方便就只讨论 $N=1$ 的情况。

记 $dp[i][j][k]$ 表示已经填了前 $i$ 列，第 $i$ 列之前最近的一个 1 离 $i$ 的距离为 $j$，已经有 $k$ 列全是 0，的方案数。那么转移就考虑下一个位置填 0 还是 1。

对于 $N=2$ 的情况则再加上一维记录第 2 行的情况。时间复杂度 $O(P^NQM)$，结合之前的算法可以得到 30 分。

# 7　40% 的算法

有 10% 的数据满足 $N \le 2$，但是 $M$ 可以达到 $10^{18}$，我们可以考虑矩阵优化递推过程。

由上一部分知动态规划，若按照已填列数划分阶段，则每一阶段的状态数是 $O(P^N Q)$ 的，因为 $N$ 比较小，所以可以手算出各种情况下的转移矩阵，再使用快速幂矩阵乘法加速递推。时间复杂度 $O((P^N Q)^3 \log M)$，期望得分 40 分。

# 8　60% 的算法

有 20% 的数据 $N$ 已经达到了 4，手算转移矩阵计算量过大，我们可以考虑用程序帮助我们找到转移矩阵。

具体做法就是，我们枚举一个状态，再枚举下一列 $N$ 个数的所有取值，算出它能转移到的状态。

时间复杂度是 $O((P^N Q)^3 \log M + P^N Q \cdot 2^N N)$，期望得分 60 分。

# 9　100% 的算法

注意到对于极限数据，若仍用之前的算法，可以发现状态数过多，矩阵乘法无法承受。

于是我们着手优化状态的表示。这里只讨论 $P = 2$ 的情况，其它的情况则是类似的。

在表示状态时，我们用 $N$ 维来表示了每一行离当前位置最近的 1 到当前位置的距离，我们称之为此时一行的 $W$ 值，注意到这个值只有 0 和 1 两种取值。而进一步注意到，我们在处理转移时，对于不同的行，只要它们的 $W$ 值相同，它们产生的转移是类似的！也就是说，真正对我们有用的，是 $W = 0$ 的行的数目和 $W = 1$ 的行的数目。于是我们可以简化状态了：$(a, b, c)$ 三元组表示 $W = 0$ 的行有 $a$ 个，$W = 1$ 的行有 $b$ 个，全是 0 的列已有 $c$ 个。

这样极大地压缩了状态。实际测试中极限数据中状态数仅有 180。

时间复杂度 $O(S^3 \log M + S \cdot 2^N N)$，其中 $S$ 是实际的状态数，期望得分 100 分。