CUCUMBER 题解

邹雨恒

1 题目大意

已知B个$N \times N$矩阵$Q_1, Q_2, \cdots, Q_B$。

对于每对$(a,b), 1 \le a < b \le B$，构造一个新的矩阵$C_{a,b}$满足：

$$C_{a,b}[i][j] = \sum_{k=1}^{N} Q_a[i][k] \times Q_b[j][k], 1 \le i, j \le N$$

对于所有$1 \sim N$的排列$p = \{p_1, p_2, \cdots, p_N\}$，定义一个新序列：

$$Tr(p)[i] = C_{a,b}[i][p_i], 1 \le i \le N$$

用$CNT(a,b)$表示使得$Tr(p)$至少有一个奇元素的排列p 的个数，定义数对(a,b) 是好的当且仅当$CNT(a,b)$ 是奇数。

计算有多少个好数对。

多组数据，$1 \le N \le 60$，$2 \le B \le 8000$，$\sum B \le 8000$。

2 算法讨论

2.1 $N = 1$的情况

在这种情况下，$C_{a,b}$仅有一个元素$Q_a[1][1] \times Q_b[1][1]$。

此时只有一个排列$p = \{1\}$。因此，(a,b)是好数对当且仅当$C_{a,b}[1][1]$是奇数，这等价于$Q_a[1][1]$和$Q_b[1][1]$都是奇数。在这种特殊情况下，我们获得了一个$O(B)$的算法。用M表示在$Q_1[1][1], Q_2[1][1], \cdots, Q_B[1][1]$中奇数的个数。那么每个数对就对应一对奇数，所以好数对的个数为$M \times (M-1)/2$。

从现在起我们假设$N \ge 2$。

2.2 简化为行列式的问题

对于给定的一个数对(a,b)我们构造一个新矩阵，使得这个矩阵行列式为奇

数当且仅当(a,b)是好数对。令$D_{a,b}[i][j]=(C_{a,b}[i][j]+1) \bmod 2$。那么对于每个排列$p=\{p_1,p_2,\cdots,p_N\}$乘积$\prod_{i=1}^{N} D_{a,b}[i][p_i]=0$当且仅当在$C_{a,b}[i][p_i]$中至少有一个奇数。如果这些数均为偶数，那么对于所有i，$D_{a,b}[i][p_i]=1$，所以乘积也为1。否则如果$C_{a,b}[i][p[i]]$ 为奇数，那么$D_{a,b}[i][p[i]]=0$，乘积也是0。

因此，这些乘积之和等于满足以下条件的排列p的个数：$Tr(p)$中元素均为偶数。也就是$N!-CNT(a,b)$。

在模2意义下，这些乘积之和等于$\det D_{a,b}$。行列式等于这些乘积之和但每一项前面带有系数1或-1，但在考虑$\det D_{a,b}$ 的时候-1和1 是相同的。

因为$N!$在$N \le 2$的时候为偶数（这也就是先考虑$N=1$的原因），我们得到$\det D_{a,b} \bmod 2 = CNT(a,b) \bmod 2$。

因此(a,b)是好数对当且仅当$\det D_{a,b} \bmod 2 = 1$。

这样我们获得了一个$O(B^2N^3)$的算法。可以使用位运算优化到$O(B^2N^2)$，但仍然太慢。

2.3 用两个矩阵的乘法表示$D_{a,b}$

注意到，如果对于所有的c,i定义$Q_c[i][N+1]=1$，就有

$$\begin{aligned} D_{a,b}[i][j] &= (C_{a,b}[i][j]+1) \bmod 2 \\ &= \left(\sum_{k=1}^{N} Q_a[i][k] \times Q_b[j][k] + 1\right) \bmod 2 \\ &= \sum_{k=1}^{N+1} Q_a[i][k] \times Q_b[j][k] \bmod 2 \end{aligned}$$

所以，我们从现在起假定Q_c是一个$N\times(N+1)$ 的矩阵，最后一行全是1。

这样我们可以将$D_{a,b}$表示为两个矩阵的乘法，也就是$D_{a,b}=Q_a^{\mathrm{T}}Q_b$。

2.4 应用Cauchy-Binet公式

利用$\det A = \det A^{\mathrm{T}}$的性质和Cauchy-Binet公式，定义$Q_{a,c}$为矩阵$Q_a$删掉第$c$列，我们有

$$\det D_{a,b} \bmod 2 = \sum_{c=1}^{N+1} \det Q_{a,c} \times \det Q_{b,c} \bmod 2$$

这样我们有了一个$O(N^3B+B^2)$的算法。预处理每个矩阵Q_a，算出所有$Q_{a,c} \bmod 2$。每个行列式可以用高斯消元和位运算在$O(N^2)$时间内算出来。因为需要$N+1$个行列式，所以时间复杂度为$O(N^3B)$。

定义

$$V_a = \sum_{c=1}^{N+1} \det Q_{a,c} \bmod 2 \times 2^{c-1}$$

可以将$\det D_{a,b}$写成$bitCount(V_a \& V_b) \bmod 2$。$bitCount(x)$表示$x$的二进制表示中1的个数，可以使用内置函数`__builtin_popcountll()` 计算。

这个算法时间复杂度中还有$O(N^3B)$的部分，仍然需要加速。

2.5 Gauss-Jordan消元法

最后一步是用$O(N^2)$的时间求出对于所有c，$\det Q_{a,c} \bmod 2$。对矩阵Q_a应用模2意义下的Gauss-Jordan消元法。也就是用初等行变换将矩阵化为行最简型。

下面假设Q_a已经成为行最简型。

首先当Q_a的秩小于N时，所有的$\det Q_{a,c} \bmod 2$均为0。当Q_a的秩等于N的时候，行最简型类似于：

```
100100
010000
001100
000010
000001
```

它是由单位矩阵中添加一列形成的。假设添加的这一列是第k列，它的第i个元素为G_i，那么对于$i \geq k$，$G_i = 0$，否则就不是行最简型。

可以得出结论：

$$\begin{aligned}
&\det Q_{a,c} \bmod 2 = G_c, 1 \leq c < k \\
&\det Q_{a,c} \bmod 2 = 1, c = k \\
&\det Q_{a,c} \bmod 2 = 0, k < c \leq N+1
\end{aligned}$$

假设$1 \leq i_1 < \cdots < i_L < k$为$G$中为1 的元素的下标，那么$G$这一列就是$i_1, i_2, \cdots, i_L$这些列之和。当$c$不是$i_1, i_2, \cdots, i_L$其中之一时，$G$与$i_1, i_2, \cdots, i_L$线性相关，所以$\det Q_{a,c} \bmod 2 = 0$。当$c = k$时，原矩阵即为单位矩阵，因此$\det Q_{a,c} \bmod 2 = 1$。当$c$在$i_1, i_2, \cdots, i_L$之中时，矩阵各列线性无关，所以$\det Q_{a,c} \bmod 2 = 1$。这样就证明了上述结论成立。

用位运算和Gauss-Jordan消元法可以在$O(N^2)$时间内完成上述过程，这样我们得到了一个$O(N^2B + B^2)$的算法，可以通过此题。

3 时空复杂度

时间复杂度：$O(N^2B + B^2)$。

空间复杂度：$O(N^2B)$。