Future of draughts(**CLOWAY**)
解题报告

CODECHEF
August Challenge 2015

杨乐（中山纪念中学）

1 题目简述

给出标号从 1 到 N 的 T 个无向无权图。在第 i 次询问中，给出三个参数 L_i, R_i 和 K_i，接着会同时在标号在 L_i 到 R_i 之间的无向图上进行游戏。

具体而言，对于标号在 L_i 到 R_i 之间的每个无向图，先**随机**地指定棋子的起始位置。然后，在每一步中，会在这些无向图中，选中一个**随机非空集合**，然后在所有被选中的无向图中，随机地将棋子移动一步。当所有无向图中棋子都回到它的起始位置时，本次询问结束。注意，在训练中至少需要进行一次行动。

对于一共的 Q 次训练，想知道有多少种可能的方案，使它在 K_i 步之内结束。由于答案可能非常大，输出它对 1000000007 取模的结果。

2 数据范围

- $1 \le T, N_k \le 50$
- $0 \le M_k \le N_k \times (N_k - 1)/2$
- $1 \le Q \le 2 \times 10^5$
- $1 \le L_i \le R_i \le T$
- $1 \le K_i \le 10^4$
- 数据集 1（10%），满足 $L_i = R_i, 1 \le K_i \le 100$
- 数据集 2（25%），满足 $1 \le K_i \le 100$
- 数据集 3（25%），满足 $1 \le K_i \le 2000, 1 \le N_k \le 15$
- 数据集 4（40%），满足 $1 \le T \le 20$
- 四组数据集互不相交。

3 解法一：容斥原理

设 $F[i][j]$ 为仅考虑第 i 张图，在上面形成长度为 j 的环的方案数。F 数组的推导可以使用简单的递推，总时间复杂度为 $O(TN^2K)$。

接下来考虑多个图合并答案的情况。考虑第 i 张图到第 j 张图，恰好 K 步结束的答案 $H[i][j][k]$。设图 X 在 K 步的过程中，有效的步数是 W（存在没有被选取的情况），这时的方案数是

$$\binom{K}{W} \times F[X][W]$$

那么图 X 贡献的总方案数为

$$G[X][K] = \sum_{i=0}^{K} \binom{K}{i} \times F[X][i]$$

直观上，答案

$$H[i][j][k] = \prod_{X=i}^{j} G[X][k]$$

但显然这样的表达式是错误的，错误在于它没有考虑到可能存在一步，这一步中所有图都没有移动，而这不符合题目的限定。所以类似于容斥原理，在总方案中去掉不合法的方案：令合法的方案数为 $P[K]$，那么递推式为：

$$P[K] = H[K] - \sum_{X=0}^{K-1} \binom{K}{X} \times P[X]$$

这样就可以在 $O(T^2K^2)$ 的时间复杂度内预处理所有可能的答案。但时间复杂度不足以通过所有的数据。

曾想过使用多项式的乘法与除法来优化处理 G 与 P 的时间复杂度，但本人尚未找到解决的方案，如果大家有解决的办法，可以与我交流细节处理问题。

4 解法二：线性代数

4.1 闭合回路：矩阵的迹 (trace)

我们先考虑只询问一个无向图的问题。用一个邻接矩阵 G 来表示图的连通情况：$G_{i,j} = 0$ 表示没有连通而 $G_{i,j} = 1$ 则代表连通。接下来记 $P(k)_{i,j}$ 为 i 与 j 之间长度为 k 的路径的数量，显然满足 $P(0)_{i,i} = 1$。而通过递推表达式

$$P(k)_{i,j} = \sum_t P(k-1)_{i,t} G_{t,j}$$

可以得到

$$P(k) = P(k-1) \cdot G = G^k$$

而长度为 k 的回路总数为 $\sum P(k)_{i,i}$，也就是矩阵对角线上的和。一个矩阵 M 的对角线上的和 $tr[M]$ 也被称为**矩阵的迹**。所以回路总数等于矩阵 G^k 的迹 $tr[G^k]$。

4.2 矩阵的特征向量 (eigenvector) 与特征值 (eigenvalue)

为了进一步了解矩阵的迹，下面来解释其在线性代数中的特性：

特征向量 (eigenvector)：一个非零列向量 v 称作矩阵 A 的一个特征向量，仅当存在 λ 使得 $Av = \lambda v$，换句话说，v 在 A 的变换中下方向不变。

特征值 (eigenvalue)：在上面的情况中，λ 称作 v 对应的特征值。让我们尝试计算一个大小为 $N \times N$ 的矩阵 A 的特征值，由定义得：

$$\begin{aligned} Av &= \lambda v \\ \lambda v - Av &= 0 \\ (\lambda I - A)v &= 0 \end{aligned}$$

其中 I 为单位矩阵。要使 v 不为 0（特征向量不能为 0），只能使矩阵 $\lambda I - A$ 的行列式为 0。从而问题变为求解方程

$$det(\lambda I - A) = 0$$

而 $det(\lambda I - A)$ 又是一个 N 次的关于 λ 的多项式，也称作该矩阵的**特征多项式 (characteristic polynomial)**。显然这个方程存在 N 个解，这些都是矩阵 A 的特征值。我们把特征值的集合称作**矩阵的谱 (spectrum)**，记作 $sp(A)$（https://en.wikipedia.org/wiki/Eigenvalues_and_eigenvectors）。

存在定理：**矩阵的迹等于它的特征值之和**。

这对我们理解问题有帮助，因为特征值存在更多的有趣的性质。例如，如果 λ 是矩阵 A 的一个特征值，那么 λ^k 是矩阵 A_k 的一个特征值（为什么？）。所以 A^k 的谱为集合 $\{\lambda^k : \lambda \in sp(A)\}$，$tr[A^k] = \sum_{\alpha \in sp(A)} \alpha^k$。

4.3 强乘积图 (strong product of two graphs) 与它的特征值

在解决一个图的情况后，我们着手于多个图结合的情况，为此引入强乘积图 (strong product of two graphs) 的定义：

在图论中，两个图 G,H 的强乘积图 $G \boxtimes H$ 是这样的一个图：

- 图中的点集是二元组 (x, y)，其中 x 来自 G，y 来自于 H
- 两个不同的顶点 (u, u') 与 (v, v') 在 $G \boxtimes H$ 中是相连的，当且仅当满足下面 3 种条件之一：
    1. u 与 v 相邻，$u' = v'$
    2. $u = v$，u' 与 v' 相邻
    3. u 与 v 相邻，u' 与 v' 相邻

根据定义，在多个图上分别计算实质上等价于在它们的**强乘积图**上计算。那么题目就转变为求一些图的强乘积图所对应的**矩阵的迹**。

下面来观察两个图的强乘积图的特征值的性质。我们可以证明，$G \boxtimes H$ 的**谱**满足

$$sp(G \boxtimes H) = \{(\alpha + 1)(\beta + 1) - 1 | \alpha \in sp(G), \beta \in sp(H)\}$$

4.4 强乘积图的迹

由上可得**强乘积图的迹**满足：

$$\begin{aligned}
tr[(G \boxtimes H)^k] &= \sum_{\alpha \in sp(G)} \sum_{\beta \in sp(H)} ((\alpha + 1)(\beta + 1) - 1)^k \\
&= \sum_{\alpha \in sp(G)} \sum_{\beta \in sp(H)} \sum_{i=0}^{k} \binom{k}{i} (\alpha + 1)^i (\beta + 1)^i (-1)^{k-i} \\
&= \sum_{i=0}^{k} \binom{k}{i} (-1)^{k-i} \sum_{\alpha \in sp(G)} (\alpha + 1)^i \sum_{\beta \in sp(H)} (\beta + 1)^i
\end{aligned}$$

又

$$tr((G + I)^k) = \sum_{\alpha \in sp(G)} (\alpha + 1)^i$$

由此可以推广到**多个图的强乘积图的迹**：

$$tr[(G_l \cdots G_r)^k] = \sum_{i=0}^{k} \binom{k}{i} (-1)^{k-i} tr[(G_l + I)^i] \cdots tr[(G_r + I)^i]$$

令

$$P(i, j, k) = \prod_{x=i}^{j} tr[(G_x + I)^k]$$

则

$$tr[(G_l \cdots G_r)^k] = \sum_{i=0}^{k} \binom{k}{i} (-1)^{k-i} P(l,r,i)$$

将整个式子化成卷积的形式，可得询问 (l,r,k) 的答案 $ans(l,r,k)$ 满足

$$ans(l,r,k) = tr[(G_l \cdots G_r)^k] = k! \sum_{i=0}^{k} \frac{P(l,r,i)}{i!} \frac{(-1)^{k-i}}{(k-i)!}$$

对于卷积这部分的操作，可以使用快速傅里叶变换 (FFT) 来优化转移时间复杂度，所以这部分的时间复杂度为 $O(T^2 K log K)$。

但由于题目所给的模数 MOD=10^9+7，并不能直接套用数论变换 (NTT)；但同时可以注意到计算后的每一项实际不大于 $K\text{MOD}^2$，所以可以使用中国剩余定理 (CRT) 处理。具体实现时，在三个不同的模数下（例如 167772161, 998244353, 1004535809）分别计算结果，接着使用中国剩余定理合并出结果。由于三个模数的乘积大于 $K\text{MOD}^2$，故能正确地求出结果。

4.5 处理 $tr[(G_i+I)^k]$ 与 $P(i,j,k)$

还剩下需要预处理的是 $tr[(G_i+I)^k]$；$P(i,j,k)$ 可以在前者处理后在 $O(T^2K)$ 的时间复杂度内计算得到。但直接计算矩阵 $(G_i+I)^k$ 的总时间复杂度为 $O(TN^3K)$，明显通过不了全部的测试数据。

方法一："大步小步"

我们若只考虑计算矩阵 $A \cdot B$ 的迹，可以在 $O(N^2)$ 的时间复杂度内计算得到，因为只需要计算目标矩阵位于对角线上的位置的结果。由此启发，可以将 $(G_i+I)^k$ 中的 k 表示为 $aP+b$，那么该矩阵则是 $(G_i+I)^{aP}$ 与 $(G_i+I)^b$ 的乘积，通过预处理所有的 $(G_i+I)^{xP}, (xP \le k)$ 与 $(G_i+I)^y, (y < P)$，可以在 $O(T(\frac{k}{P}+p)N^3 + TN^2K)$ 时间复杂度内计算得到。明显当 $P=\sqrt{K}$ 时总时间复杂度最小，可以通过全部测试数据。

方法二：特征多项式与 Cayley-Hamilton theorem

回到线性代数上，我们有更优美（复杂）的理论与算法。

为了方便叙述，令 $H=G+I$，我们的计算目标是 $tr[I], tr[H], tr[H^2], \ldots, tr[H^k]$。

由 Cayley-Hamilton theorem 可以得到，如果 $c_0+c_1x+c_2x^2+\cdots+x^N$ 是矩阵 H 的特征多项式，则 $c_0I+c_1H+c_2H^2+\cdots+H^N=0$（注意 0 指的是一个零矩阵而非数字 0）。

所以，我们可以推导得到（在等式两端同时乘以 H^{k-N}）：

$$\begin{aligned} c_0I + c_1H + c_2H^2 + \cdots + H^N &= 0 \\ c_0H^{k-N} + c_1H^{k-N+1} + c_2H^{k-N+2} + \cdots + H^k &= 0 \\ tr[c_0H^{k-N} + c_1H^{k-N+1} + c_2H^{k-N+2} + \cdots + H^k] &= 0 \\ c_0tr[H^{k-N}] + c_1tr[H^{k-N+1}] + c_2tr[H^{k-N+2}] + \cdots + tr[H^k] &= 0 \end{aligned}$$

最后一步的推导利用了**矩阵的迹**是**线性映射（linear map）**的性质。

由此，我们只需暴力计算出 $tr[I], tr[H], tr[H^2], \ldots, tr[H^N]$，再求出 H 的特征多项式，使用递推来计算出 $tr[H^(N+1)], \ldots, tr[H^k]$。所以这部分的总时间复杂度为 $O(TN^4 + TNK)$。

计算特征多项式（characteristic polynomial）

要使用上述的方法前提是求出某一个矩阵 H 的特征多项式。根据定义，它等于 $det(xI-H)$，是一个 N 次的**首一多项式（monic polynomial）**(首项为 1)，它们的根的集合就是**矩阵 H 的谱**。假设矩阵 H 的特征值分别为 $\lambda_1, \lambda_2, \ldots, \lambda_N$，那么它的特征多项式就是 $(x-\lambda_1)(x-\lambda_2)\cdots(x-\lambda_N)$。

举个例子，假设 $N = 3$，那么它的特征多项式就像这个形式：

$$(x-\lambda_1)(x-\lambda_2)(x-\lambda_3) = x^3-(\lambda_1+\lambda_2+\lambda_3)x^2+(\lambda_1\lambda_2+\lambda_1\lambda_3+\lambda_2\lambda_3)x-\lambda_1\lambda_2\lambda_3$$

$(x_1 + x_2 + x_3), (x_1x_2 + x_2x_3 + x_1x_3), (x_1x_2x_3)$ 称作 x_1, x_2, x_3 这三个变量的**初等对称多项式（elementary symmetric polynomial）**，这些多项式与一个多项式的系数与根有着很大的联系。

k 阶初等对称多项式（elementary symmetric polynomial of order k）记作 $e_k(x_1, \ldots, x_N)$，它是所有的在 $\{x_1, \ldots, x_N\}$ 中大小为 k 的子集的乘积之和。如果将集合中的数赋值为特征多项式的根（即特征值），那么，显然地，特征多项式的系数满足：

$$\begin{gathered} e_1(\lambda_1, \ldots, \lambda_N) = -c_{N-1} \\ e_2(\lambda_1, \ldots, \lambda_N) = +c_{N-2} \\ e_3(\lambda_1, \ldots, \lambda_N) = -c_{N-3} \\ e_4(\lambda_1, \ldots, \lambda_N) = +c_{N-4} \\ \cdots \\ e_N(\lambda_1, \ldots, \lambda_N) = (-1)^N c_0 \end{gathered}$$

那么，我们希望计算 $e_i(\lambda_1, \ldots, \lambda_N)$。

计算的关键在于使用 $tr[I], tr[H], tr[H^2], \ldots, tr[H^N]$（已经在前面计算过了）。由于 $tr[H^k]$ 等于所有特征值的 k 次方之和，所以运用**牛顿恒等式**

（**Newton's identities**）可以推导出：

$$ke_k(x_1,\ldots,x_N) = \sum i = 1^k(-1)^{i-1}e_{k-i}(x_1,\ldots,x_N)p_i(x_1,\ldots,x_N)$$

其中，$p_i(x_1,\ldots,x_N)$ 被定义为 $x_1^i + x_2^i + \cdots + x_N^i$，正是 $tr[H^i]$。
所以递推计算 e 的时间复杂度为 $O(N^2)$，是十分优秀的。

5 归纳总结

本题是一道冗长的线性代数题目，需要缜密的分析与清晰的实现思路：

1. 对于每一个无向图 G，令 $H = G + I$，计算出它的 $tr[H^i](0 \le i \le k)$。
2. 计算出 $P(i, j, k)$。
3. 计算出 $ans(i, j, k)$。

对于第一步，可以使用“大步小步”法或特征多项式法；
对于第三步，要使用 FFT 优化卷积的计算。
总时间复杂度为 $O(TN^4 + TNK + T^2KlogK + Q)$；
而空间复杂度为 $O(TNK)$。

解法二：线性代数主要参考的是在 Codechef上的官方题解（https://discuss.codechef.com/questions/73733/cloway-editorial），并在其上加以归纳总结，也省略了一些无关紧要部分。建议对照中英文阅读能更好理解线性代数中的定义与解题方法。