

I题题解

(这是一道模板题)

- 如果使用B题的矩阵乘法进行暴力连乘，时间复杂度为 $O(mn^3)$ ，显然会超时，所以需要使用矩阵快速幂进行优化。
- 矩阵快速幂的原理和快速幂相同，都是将指数不断对半拆分，然后通过矩阵乘法进行快速计算。

以下是示例代码：

```
#include <stdio.h>
#include<math.h>
#include<string.h>
#include<ctype.h>
#include<stdlib.h>
#include<stdbool.h>
#define MOD 1000000007
//矩阵乘法
void multiply(int n, long long A[128][128], long long B[128][128], long long AB[128][128]) {
    long long temp[128][128] = {0};
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++) {
            temp[i][j] = 0;
            for (int k = 0; k < n; k++) {
                temp[i][j] = (temp[i][j] + A[i][k] * B[k][j]) % MOD;
            }
        }
    }
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++) {
            AB[i][j] = temp[i][j];
        }
    }
}
//矩阵快速幂
void matrix_pow(int n, long long A[128][128], long long m, long long res[128][128]) {
    long long temp[128][128];
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++) {
            res[i][j] = (i == j);
        }
    }
    while (m > 0) {
        if (m % 2 == 1) {
            multiply(n, res, A, res);
        }
        multiply(n, A, A, A);
        m /= 2;
    }
}
```

```
}
int main() {
    int t;
    scanf("%d", &t);
    while (t--) {
        int n,m;
        scanf("%d %d", &n, &m);
        long long A[128][128], B[128][128], C[128][128], AB[128][128];
        for (int i = 0; i < n; i++) {
            for (int j = 0; j < n; j++) {
                scanf("%lld", &A[i][j]);
            }
        }
        for (int i = 0; i < n; i++) {
            for (int j = 0; j < n; j++) {
                scanf("%lld", &B[i][j]);
            }
        }
        multiply(n, A, B, AB);
        matrix_pow(n, AB, m, C);
        for (int i = 0; i < n; i++) {
            for (int j = 0; j < n; j++) {
                printf("%lld ", C[i][j]);
            }
            printf("\n");
        }
    }
    return 0;
}
```