

C2-B题题解

题目描述

给定两个n阶整数方阵A, B, 试求A×B。

题目分析

本题是一个n阶整数方阵求乘积的问题, 考虑到矩阵运算法则和本题数据规模, 可以采用分治法解决问题。利用二维数组来存储第i行第j列的数据。由矩阵算法可知, 每计算一个C[i][j], 需要做n次乘法和n-1次加法, 因此计算C的n*n个元素需要n^3次乘法和n^3-n^2次加法, 因此算法时间复杂度为O (n^3) 。具体代码如下。

详细代码

```
#include <stdio.h>
#include <string.h>

int main(void) {
    int K;
    scanf("%d\n", &K);
    while (K--) {
        int T;
        scanf("%d", &T);
        long long a[T][T] = {0}, b[T][T] = {0}, c[T][T] = {0}, d[T][T] = {0};
        \\创建二维数组, 注意采用long long数据类型。
        long long i, j, sum = 0, k;
        for (i = 0; i < T; i++) {
            for (j = 0; j < T; j++) {
                scanf("%lld", &a[i][j]); \\获取初始矩阵a
            }
        }
        for (i = 0; i < T; i++) {
            for (j = 0; j < T; j++) {
                scanf("%lld", &b[i][j]); \\获取初始矩阵b
            }
        }

        for (i = 0; i < T; i++) {
            for (j = 0; j < T; j++) {
                sum = 0;
                for (k = 0; k < T; k++) {
                    sum = sum + a[i][k] * b[k][j]; \\依次计算
                }
                d[i][j] = sum;
            }
        }

        for (i = 0; i < T; i++) {
            for (j = 0; j < T; j++) {
                printf("%lld ", d[i][j]); \\输出结果
            }
        }
    }
}
```

```
        }  
        printf("\n");  
    }  
}  
return 0;  
  
}
```