

L1530 : 二维数组前缀和 ☆☆

题目描述

给定一个 n 行 m 列的二维数组，你需要计算该数组的前缀和。

所谓前缀和是指：对于每个位置 (i, j) ，求出从左上角 $(0, 0)$ 到当前位置 (i, j) 的所有元素的和。

$$a = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \Rightarrow s = \begin{bmatrix} 1 & 3 & 6 \\ 5 & 12 & 21 \\ 12 & 27 & 45 \end{bmatrix}$$

输入输出格式

输入：第一行包含两个整数 n, m ($1 \leq n, m \leq 1000$)，表示二维数组的行数和列数。

接下来的 n 行，每行包含 m 个整数，表示二维数组的元素。

每个元素的值满足 $-10^3 \leq a[i][j] \leq 10^3$ 。

输出：输出 n 行，每行包含 m 个整数，表示前缀和数组对应位置的值。

每行数之间用一个空格分隔，且每行末尾不要有空格。

输入示例	输出示例
3 3	1 3 6
1 2 3	5 12 21
4 5 6	12 27 45
7 8 9	

样例解释

以计算 $s[1][2]$ (第二行第三列) 为例，演示二维前缀和的计算过程：

$$\begin{aligned}s[1][2] &= a[0][0] + a[0][1] + a[0][2] + a[1][0] + a[1][1] + a[1][2] \\&= 1 + 2 + 3 + 4 + 5 + 6 \\&= 21\end{aligned}$$

算法分析

1. 直接计算方法

对于每个位置 (i, j) ，直接计算从 $(0, 0)$ 到 (i, j) 的所有元素之和：

```
for(int x = 0; x <= i; x++) {
    for(int y = 0; y <= j; y++) {
        s[i][j] += a[x][y];
    }
}
```

这种方法的时间复杂度为 $O(n^2 \cdot m^2)$ ，会导致超时。

2. 优化方法：前缀和

利用已计算的前缀和来计算新的前缀和，可以大大降低时间复杂度。以下以 $s[2][2]$ 的计算为例进行说明。

(1) 区域划分

我们用不同颜色表示不同的前缀和区域， $s[2][2] =$ 整个矩形内所有数字的和：

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \quad \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \quad \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

- 蓝色部分的数字和表示 $s[1][2]$ ，红色部分的数字和表示 $s[2][1]$ ，绿色部分的数字和表示 $s[1][1]$ （红蓝重叠的部分），橘色部分的数字表示 $a[2][2]$ 。
- 整个矩形的和 = 蓝色区域 + 红色区域 - 绿色区域 + 橘色区域
- 用符号表示就是： $s[2][2] = s[1][2] + s[2][1] - s[1][1] + a[2][2]$
- 得出通用公式： $s[x][y] = s[x-1][y] + s[x][y-1] - s[x-1][y-1] + a[x][y]$
当前位置前缀和 = 左前缀和 + 上前缀和 - 左上角前缀和 + 当前位置的值
- 我们利用这个公式，从左上角到右下角的顺序，计算 $s[i][j]$ ，就可以得到所有 $s[x][y]$ 的值。

代码实现：

处理前缀和

```
for (int i = 1; i <= n; i++) {
    for (int j = 1; j <= m; j++) {
        s[i][j] += s[i - 1][j] + s[i][j - 1] -
                    s[i - 1][j - 1] + a[i - 1][j - 1];
    }
}
```

输出格式

```
//注意行末不能有多余的空格
for (int i = 0; i < n; i++) {
    for (int j = 0; j < m; j++) {
        cout << a[i][j] << " \n"[j == m - 1];
    }
}
```

拓展思考

- 如果换成三维，如何求 $a[i][j][k]$ 的前缀和？
- 如果要求从 (x_1, y_1) 到 (x_2, y_2) 的区域和，该如何计算？