前缀和

TEACHING COURSEWARE POWPOINT

授课时间：2025.07.28

目录

PART-01 概念引入 TEACHING COURSEWARE

PART-02 概念分析 TEACHING COURSEWARE

PART-03 经典例题 TEACHING COURSEWARE

PART-04 教学反思 TEACHING COURSEWARE

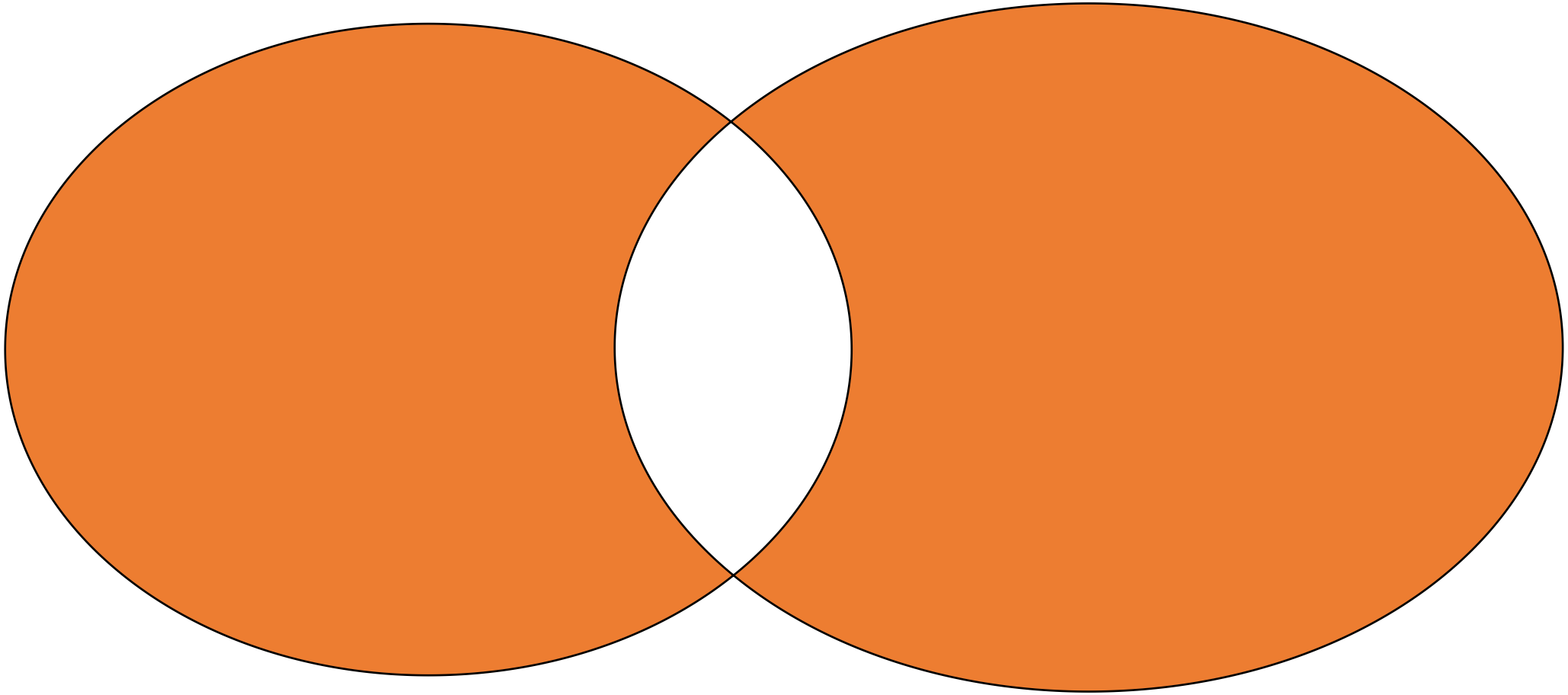
01

概念引入

TEACHING
COURSEWARE

TEACH

容斥原理



02

概念分析

TEACHING
COURSEWARE

TEACH

概念分析

二维前缀和是一种用于高效计算二维矩阵中任意子矩形区域元素和的预处理技术。它的核心思想是通过预先计算一个“前缀和矩阵”，将原本需要遍历子矩阵才能得到的和，转化为通过前缀和矩阵的简单加减运算即可快速求得，从而将单次查询的时间复杂度从 O（rows×cols）降至 O（1）。

概念分析

二维前缀和的计算公式为：
pre[i][j] = a[i][j] + pre[i-1][j] + pre[i][j-1] - pre[i-1][j-1]
a[i][j]：当前位置（i, j）本身的元素值。
pre[i-1][j]：上方矩形的前缀和，即从（1, 1）到（i-1, j）的所有元素之和（当前位置正上方的区域）。
pre[i][j-1]：左方矩形的前缀和，即从（1, 1）到（i, j-1）的所有元素之和（当前位置正左方的区域）。
pre[i-1][j-1]：左上角重叠矩形的前缀和，即 pre[i-1][j] 和 pre[i][j-1] 重叠的区域（从（1, 1）到（i-1, j-1））。

假设需要计算子矩阵（x1, y1）到（x2, y2）（左上角到右下角）的元素和，利用前缀和矩阵可以通过以下公式直接求得：
sum = pre[x2][y2] - pre[x1-1][y2] - pre[x2][y1-1] + pre[x1-1][y1-1]

03

经典例题

TEACHING
COURSEWARE

TEACH

经典例题

题目描述

给定一个 n 行 m 列的整数矩阵，随后有 q 次查询。

每次查询会给出两个坐标 (x_1, y_1) 和 (x_2, y_2)，
表示一个矩 形区域的左上角和右下角

请你计算并输出该矩形区域内所有元素的总和。

输出格式

对于每次查询，输出对应矩形区域的元素总和，每个结果占一行。

输入格式

第一行包含两个整数 n 和 m，分别表示矩阵的行数和列数。

接下来 n 行，每行包含 m 个整数，表示矩阵的元素。

随后一行包含一个整数 q，表示查询的次数。

接下来 q 行，每行包含四个整数 x_1, y_1, x_2, y_2，表示一次查询的矩形区域。

经典例题

输入样例：	输出样例：
3 3 1 2 3 4 5 6 7 8 9 3 1 1 2 2 1 3 3 3 2 1 3 2	12 18 24

标准程序


```cpp
#include <iostream>
using namespace std;
const int N = 1010;
int a[N][N];          // 存储原始矩阵，下标从1开始
long long pre[N][N]; // 存储二维前缀和，下标从1开始，避免边界判断
int main() {
    int n, m;
    cin >> n >> m;
    for (int i = 1; i <= n; ++i) {
        for (int j = 1; j <= m; ++j) {
            cin >> a[i][j];
            pre[i][j] = a[i][j] + pre[i-1][j] + pre[i][j-1] - pre[i-1][j-1];
        }
    }
```

标准程序


```
int q;
cin >> q;
while (q--) {
    int x1, y1, x2, y2;
    cin >> x1 >> y1 >> x2 >> y2;
    // 计算并输出矩形区域和
    long long sum = pre[x2][y2] - pre[x1-1][y2] - pre[x2][y1-1] + pre[x1-1][y1-1];
    cout << sum << '\n';
}

return 0;
```

04

Q&A

TEACHING
COURSEWARE

TEACH

Q&A

Q:什么是二维前缀和？它的核心作用是什么？

Q:计算二维前缀和矩阵pre[i][j]的公式是什么？