```cpp
#include "../bits/stdc++.h"
// 2次元BIT, 1-indexed
// 2次元累積和クエリ O(logMlogN)
// verified: https://atcoder.jp/contests/joi2011ho/submissions/4247715
class FenwickTree2D
{
    // 2次元(M*N) 1-indexed BIT
    int M, N;
    // bit[x][y] = v[i][j] (Lx < i <= x, Ly < j <= y)
    // Lx = x - (x & -x), Ly = y - (x & -y)
    std::vector<std::vector<int>> bit;

  public:
    FenwickTree2D(int _m, int _n) : M(_m), N(_n), bit(M + 1, std::vector<int>(N + 1, 0)) {}
    // v[a][b] += w
    void add(int a, int b, int w)
    {
        for (int x = a; x <= M; x += x & -x)
            for (int y = b; y <= N; y += y & -y)
                bit[x][y] += w;
    }
    // O(logM logN)
    int sum(int a, int b)
    {
        int ret = 0;
        for (int x = a; x > 0; x -= x & -x)
            for (int y = b; y > 0; y -= y & -y)
                ret += bit[x][y];
        return ret;
    }
    // (i,j) (a <= i <= c,b <= j <= d) の長方形領域内の和
    int sum(int a, int b, int c, int d)
    {
        int ret = sum(c, d);
        ret -= sum(c, b - 1);
        ret -= sum(a - 1, d);
        ret += sum(a - 1, b - 1);
        return ret;
    }
};

```