FenwickTree (BIT) sum query on 2D-Array:

```cpp
//range-update and range-query //One-based indexing
struct FenwickTree2D{
    vector<vector<vector<ll>>> M , A;
    const int n , m;

    FenwickTree2D(int n , int m) : n(n) , m(m) {
        M.assign(n, vector<vector<ll>>(m , vector<ll>(2 , 0)));
        A.assign(n, vector<vector<ll>>(m , vector<ll>(2 , 0)));
    }

    FenwickTree2D(vector<vector<ll>> a) : FenwickTree2D(a.size() , a[0].size()){
        for (int i = 1; i < a.size(); i++)
            for (int j = 1; j < a[0].size(); j++)
                update(i , j , i , j , a[i][j]);
    }

    void upd2(vector<vector<vector<ll>>> &t, int x, int y, ll mul, ll add){
        for (int i = x; i < n; i += i & -i){
            for (int j = y; j < m; j += j & -j){
                t[i][j][0] += mul;
                t[i][j][1] += add;
            }
        }
    }

    void upd1(int x, int y1, int y2, ll mul, ll add){
        upd2(M, x, y1, mul, -mul * (y1 - 1));
        upd2(M, x, y2, -mul, mul * y2);
        upd2(A, x, y1, add, -add * (y1 - 1));
        upd2(A, x, y2, -add, add * y2);
    }

    void update(int x1, int y1, int x2, int y2, ll val){
        upd1(x1, y1, y2, val, -val * (x1 - 1));
        upd1(x2, y1, y2, -val, val * x2);
    }

    ll query2(vector<vector<vector<ll>>> &t , int x, int y){
        ll mul = 0, add = 0;
        for (int i = y; i > 0; i -= i & -i){
            mul += t[x][i][0];
            add += t[x][i][1];
        }
        return mul * y + add;
    }

    ll query1(int x, int y){
        ll mul = 0, add = 0;
        for (int i = x; i > 0; i -= i & -i){
            mul += query2(M, i, y);
            add += query2(A, i, y);
        }
        return mul * x + add;
    }

    ll query(int x1, int y1, int x2, int y2){
        return query1(x2, y2)-query1(x1-1, y2)-query1(x2, y1-1)+query1(x1-1, y1-1);
    }
};
```