



**Факультет программной инженерии и
компьютерной техники**

Алгоритмы и структуры данных

Лабораторная работа №5

Структуры данных

Преподаватель: Косяков Михаил Сергеевич

Выполнил: Ле Чонг Дат

Группа: P3231

2021 г.

1 Задание

Вариант: 1628. White Streaks

The life of every unlucky person has not only black but also white streaks. The Martian Vas-Vas has a calendar in the form of an $m \times n$ table; he marks in this calendar days when he had bad luck. If Vas-Vas had bad luck in the j th day of the i th week, he paints the cell (i, j) black. Initially, all cells are white. Let rectangles of the form $1 \times l$ or $l \times 1$ be called segments of life. Maximal with respect to inclusion white segments are called white streaks. Can you determine how many white streaks there were in the life of Vas-Vas?

2 Решение

Мы будем использовать вектор-столбец и вектор-строку для хранения местоположения заблокированных ячеек.

Легко видеть, что если длина ячеек между двумя заблокированными ячейками больше 1, к результату будет добавлена 1.

В случае, если длина ячеек между 2 заблокированными ячейками равна 1, мы проверим, принадлежит ли ячейка другому сегменту с длиной больше 1, если не добавим результат к 1.

3 Реализация

```
#include <bits/stdc++.h>
using namespace std;
const int N = 300010;
int ans;
int m, n, k;
vector<int> row[N], col[N];
map<pair<int, int>, bool> blocked;
int main() {
    ios::sync_with_stdio(0);
    cin.tie(0); cout.tie(0);
    cin >> m >> n >> k;
    for(int i = 1; i <= m; ++i) {
        row[i].emplace_back(0);
        row[i].emplace_back(n+1);
        blocked[{i, 0}] = true;
        blocked[{i, n+1}] = true;
    }
    for(int i = 1; i <= n; ++i) {
        col[i].emplace_back(0);
        col[i].emplace_back(m+1);
        blocked[{0, i}] = true;
        blocked[{m+1, i}] = true;
    }
    for(int i = 1; i <= k; ++i) {
        int u, v;
```

```

        cin >> u >> v;
        row[u].emplace_back(v);
        col[v].emplace_back(u);
        blocked[{u, v}] = true;
    }
    for(int i = 1; i <= m; ++i) {
        sort(row[i].begin(), row[i].end());
        for(int j = 0; j < (int) row[i].size() - 1; ++j) {
            int amount = row[i][j+1] - row[i][j] - 1;
            ans += (amount > 1);
        }
    }
    for(int i = 1; i <= n; ++i) {
        sort(col[i].begin(), col[i].end());
        for(int j = 0; j < (int) col[i].size() - 1; ++j) {
            int amount = col[i][j+1] - col[i][j] - 1;
            switch(amount) {
                case 0:
                    break;
                case 1:
                    ans += (blocked.count({col[i][j+1]-1, i-1}) and
                        blocked.count({col[i][j+1]-1, i+1}));
                    break;
                default:
                    ans ++;
            }
        }
    }
    cout << ans;
}

```

4 Вывод

Я научился использовать векторные структуры данных. Я научился использовать vector и map структуры данных.