

Санкт-Петербургский национальный исследовательский университет ИТМО

Факультет систем управления и робототехники

Алгоритмы и структуры данных

Отчёт по лабораторной работе №14 (1628. Белые полосы)

Преподаватель: Тропченко А. А.

Выполнил: Марухленко Д. С.

Группа: R3235

Санкт Петербург, 2021г.

1. Цель работы

Решить задачу №1628 на платформе Timus Online Judge

<https://acm.timus.ru/problem.aspx?space=1&num=1628>

2. Задача

Условие

У каждого неудачника в жизни бывают не только чёрные, но и белые полосы. Марсианин Вась-Вась отмечает в календаре, представляющем собой таблицу $m \times n$, те дни, когда ему ужасно не повезло. Если Вась-Васю не повезло в j -й день i -й недели, то он закрашивает ячейку таблицы (i, j) в чёрный цвет. Все не закрашенные ячейки в таблице имеют белый цвет.

Будем называть отрезками жизни прямоугольники размером 1×1 либо 1×1 . Белыми полосами Вась-Вась считает все максимальные по включению белые отрезки таблицы. А сможете ли Вы определить, сколько всего белых полос было в жизни Вась-Вася?

Ограничение времени: 1.0 секунды

Ограничение памяти: 64 МБ

Исходные данные

Первая строка содержит целые числа m, n, k — размеры календаря и количество неудачных дней в жизни Вась-Вася ($1 \leq m, n \leq 30000$; $0 \leq k \leq 60000$). В следующих k строках перечислены неудачные дни в виде пар (x_i, y_i) , где x_i — номер недели, к которой относится неудачный день, а y_i — номер дня в этой неделе ($1 \leq x_i \leq m$; $1 \leq y_i \leq n$). Описание каждого неудачного дня встречается только один раз.

Результат

Выведите число белых полос в жизни Вась-Вася.

Пример

Исходные данные	Результат
3 5 4 1 1 1 5 2 2 3 3	8
5 1 2 2 1 3 1	2

3. Материалы работы

3.1. Объяснение алгоритма

Для решения задачи создаётся вектор, в котором хранятся все «чёрные» дни. Так же для разделения строк и столбцов по границам календаря создаются дополнительные «чёрные» дни, после чего вектор сортируется по координатам, тем самым вытягивая таблицу по строкам. Далее происходит проверка расстояния между «чёрными» днями и подсчёт полос. Если длина полосы равна 1, то эта клетка добавляется в список клеток для повторной проверки(не является ли клетка частью другой полосы). После – проводится сортировка по столбцам и проводятся аналогичные операции с повторной проверкой линий длины 1. В конце выводится количество найденных полос

3.2. Код программы.

```
1. #include <algorithm>
2. #include <iostream>
3. #include <vector>
4. #include <set>
5. using namespace std;
6.
7. bool compare_y(pair<int, int> a, pair<int, int> b) {
8.     if (a.second != b.second) return a.second < b.second;
9.     return a.first < b.first;
10. }
11.
12. bool compare_x(pair<int, int> a, pair<int, int> b) {
13.     if (a.first != b.first) return a.first < b.first;
14.     return a.second < b.second;
15. }
16.
17. int counter;
18. int main() {
19.     int N, M, K;
20.
21.     cin >> N >> M >> K;
22.     vector<pair<int, int>> cell;
23.     set<pair<int, int> > S;
24.     int x, y;
25.     for (int i = 0; i < K; ++i) {
26.         cin >> x >> y;
27.         cell.push_back(make_pair(x, y));
28.     }
29.
30.     for (int i = 1; i <= M; i++) {
31.         cell.push_back(make_pair(0, i));
32.         cell.push_back(make_pair(N + 1, i));
33.     }
34.
35.     for (int i = 1; i <= N; i++) {
36.         cell.push_back(make_pair(i, 0));
37.         cell.push_back(make_pair(i, M + 1));
38.     }
39.
40.     K += (M + N) * 2;
41.
42.     sort(cell.begin(), cell.begin() + K, compare_y);
43.     for (int i = 0; i < K; i++) {
44.         int space = cell[i + 1].first - cell[i].first;
45.         if (space == 2) {
46.             if (S.find(make_pair(cell[i].first + 1, cell[i].second)) != S.end())
47.                 counter++;
48.         }
49.         else if (cell[i].second == cell[i + 1].second && space > 2)
50.             counter++;
```

```

51.     }
52.
53.     sort(cell.begin(), cell.begin() + K, compare_x);
54.     for (int i = 0; i < K ; i++) {
55.         int space = cell[i + 1].second - cell[i].second;
56.         if (space == 2)
57.             S.insert(make_pair(cell[i].first, cell[i].second + 1));
58.         else if (cell[i].first == cell[i + 1].first && space > 2)
59.             counter ++;
60.     }
61.
62.     cout << counter << endl;
63.     return 0;
64. }

```

65. Результат выполнения и ссылка на репозиторий GitHub

ID	Дата	Автор	Задача	Язык	Результат проверки	№ теста	Время работы	Выделено памяти
9328818	14:19:09 19 апр 2021	Daniil Marukhlenko	1628	G++ 9.2 x64	Accepted		0.187	4 216 КБ

https://github.com/japersik/algorithms_and_data_structures/



4. Вывод

Работа выполнена, задача решена с использованием сортировок и структур данных.