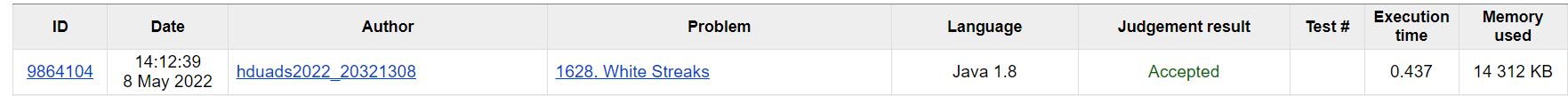
Laboratory work #6

Student: CAO Xinyang  
Student ID: 20321308  
Timus Name: hduads2022\_20321308

Mail: c.x\_yang@foxmail.com

Problem #1628

Screenshot from Timus:



Explanation of algorithm:

Create an array of size (m+2) \* (n+2) to store black and white, surrounded by black cells. Read the length of White Streaks row by column. If the length is greater than 1; If the length is equal to 1, it determines whether this is a 1\*1 cell, and if so, it counts to the total

Computational complexity of algorithm:

T(N) = O(m\*n)

Source code:

import java.io.\*;

public class App {

public static void main(String[] args) throws Exception {

StreamTokenizer in = new StreamTokenizer(new BufferedReader(new InputStreamReader(System.in)));

PrintWriter out = new PrintWriter(System.out);

in.nextToken();

int m = (int) in.nval;

in.nextToken();

int n = (int) in.nval;

in.nextToken();

int k = (int) in.nval;

boolean[][] calendar = new boolean[m + 2][n + 2];

for (int i = 0; i <= m + 1; i++) {

calendar[i][0] = true;

calendar[i][n + 1] = true;

}

for (int i = 1; i <= n; i++) {

calendar[0][i] = true;

calendar[m + 1][i] = true;

}

for (int i = 0; i < k; i++) {

in.nextToken();

int x = (int) in.nval;

in.nextToken();

int y = (int) in.nval;

calendar[x][y] = true;

}

int ans = 0;

// vertical

for (int i = 1; i <= m; i++) {

int cur = 0;

for (int j = 1; j <= n + 1; j++) {

if (calendar[i][j]) {

int length = j - cur - 1;

if (length > 1) {

ans++;

} else if (length == 1) {

if (calendar[i - 1][cur + 1] && calendar[i + 1][cur + 1])

ans++;

}

cur = j;

}

}

}

// horizontal

for (int j = 1; j <= n; j++) {

int cur = 0;

for (int i = 1; i <= m + 1; i++) {

if (calendar[i][j]) {

if (i - cur - 1 > 1) {

ans++;

}

cur = i;

}

}

}

out.println(ans);

out.flush();

}

}