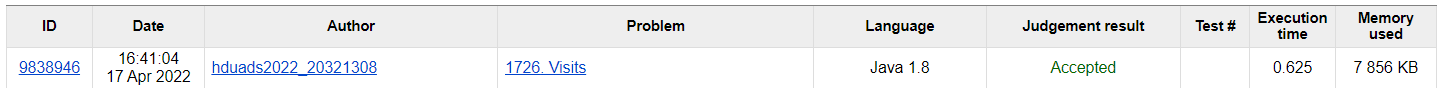
Laboratory work #4

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

Mail: c.x\_yang@foxmail.com

Problem #1726

Screenshot from Timus:



Explanation of algorithm:

Add up all the possible paths and find the average. To reduce the computation, first sort the arrays x and y, then sum and average the distances. And the particular thing to pay attention to is the range of x, y and n. To prevent an array overflow, the Long data type is used.

Computational complexity of algorithm:

F(N) = N - 1

T(N) = O(N)

Source code:

import java.util.Scanner;

import java.util.Arrays;

public class App {

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

Scanner scan = new Scanner(System.in);

long n = scan.nextLong();

long[] x = new long[(int)n];

long[] y = new long[(int)n];

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

x[i] = scan.nextInt();

y[i] = scan.nextInt();

}

Arrays.sort(x);

Arrays.sort(y);

long ans = 0;

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

long temp = (x[i] - x[i - 1] + y[i] - y[i - 1]);

temp \*= i \* (n - i) \* 2;

ans += temp;

}

ans /= n \* (n - 1);

System.out.println(ans);

scan.close();

}

}