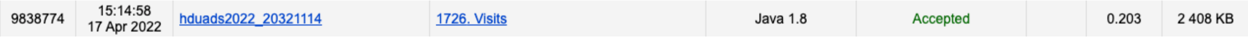
Laboratory work # 4

Student: HU Riqian
Student ID: 20321114
Timus Name: hduads2022_20321114

Mail: jhlxhrq@163.com

Problem # 1726. Visits

Screenshot from Timus:



9838774	15:14:58 17 Apr 2022	hduads2022_20321114	1726. Visits	Java 1.8	Accepted		0.203	2 408 KB

Explanation of algorithm:

1. I use the long[] X_AXIS and the long[] Y_AXIS to store the corresponding points on the plane;

2. For the points on the plane, I add up their x_axis_projection and y_axis_projection desperately;

3. I firstly sort the array, then in the for-loop I add them up.

Computational complexity of algorithm:

$$O(2N)$$

Source code:

```
import java.io.*;
import java.util.Arrays;

public class Visit {
    public static void main(String[] args) throws IOException {
        new Visit().run();
    }
    StreamTokenizer in;
    PrintWriter out;
    static long SIZE;
    static long[] X_AXIS;
    static long[] Y_AXIS;

    int nextInt() throws IOException {
        in.nextToken();
        return (int)in.nval;
    }

    void run() throws IOException {
        in = new StreamTokenizer(new BufferedReader(new
InputStreamReader(System.in)));
```

```
        out = new PrintWriter(System.out);
        inputData();
        int answer = solve();
        out.println(answer);
        out.flush();
    }

    void inputData() throws IOException {
        SIZE = nextInt();
        X_AXIS = new long[(int) SIZE];
        Y_AXIS = new long[(int) SIZE];
        for (int i = 0; i < SIZE; i++) {
            X_AXIS[i] = nextInt();
            Y_AXIS[i] = nextInt();
        }
    }

    int solve() throws IOException {
        long totalVisits = SIZE * (SIZE - 1) / 2;
        double totalPath = solveVisits(X_AXIS) + solveVisits(Y_AXIS);
        return (int) Math.floor(totalPath / totalVisits);
    }

    long solveVisits(long[] array) {
        Arrays.sort(array);
        long totalPath = 0;
        for (long i = 1; i < SIZE; i++) {
            totalPath += Math.abs(array[(int) i] - array[(int) (i - 1)]) * i
* (SIZE - i);
        }
        return totalPath;
    }
}
```