[CS 11] Prac 1c – Darts II: Electric Boogaloo

oj.dcs.upd.edu.ph/problem/cs11prac1c

Problem Statement

Steve has been practicing darts a lot!

In Steve's world, the dartboard is an **axis-aligned rectangle**, that is, a rectangle each of whose sides is parallel to either the x-axis or y-axis.

The dartboard can be represented as the axis-aligned rectangle in the 2D Cartesian plane whose bottomleft corner has coordinates (x1,y1) (x_1 , y_1) and topright corner has coordinates (x2,y2) (x_2 , y_2) .

Steve has thrown several darts so far. We know the coordinates of where each dart landed.

How many of Steve's darts landed inside the dartboard?

Assume that a dart is a point, and that the boundary of the rectangle counts as being *inside* the rectangle. It is also allowed for multiple darts to hit the same point on the plane.

Task Details

Your task is to implement a function called `darts_landed`. This function has five parameters `x1`, `y1`, `x2`, `y2`, `dart_coords` in that order.

- `x1`, `y1`, `x2` and `y2` are all `int`s and their meanings are described in the problem statement.
- `dart_coords` is a tuple of pairs (x,y) (x , y) , each of which denotes the coordinates of a point where a dart landed.

The function must return an `int` denoting the number of darts that landed inside the dartboard.

Recall that a **pair** is just a 22-tuple.

Restrictions

For this problem:

- Assignment is allowed.

- Recursion is allowed.
- Up to 66 function definitions are allowed.
- Comprehensions are **disallowed**.
- `range` is **disallowed**.
- The `abs` symbol is now allowed.
- The source code limit is 10001000.

Example Calls

Example 1 Function Call

Copy

```
darts_landed(-20, -10, 20, 10, (
    (0, 0),
    (500, 0),
    (1, 1),
))
```

Example 1 Return Value

Copy

```
2
```

Constraints

- The function `darts_landed` will be called at most 1010 times.
- Each argument will have an absolute value at most 102010^{20}.
- It is guaranteed that x1<x2$x_1 < x_2$ and y1<y2$y_1 < y_2$.
- The length of `dart_coords` is at most 100100.

Scoring

You get 100100 ❤️ points if you solve all test cases correctly.
Report an issue

Clarifications

No clarifications have been made at this time.