1h 8m left

ALL

1. Nearly Similar Rectangles

Recently, while researching about similar rectangles, you found the term "Nearly Similar Rectangle." Two rectangles with sides *(a, b)* and *(c, d)* are nearly similar only if *a/c = b/d*. The order of sides matter in this definition, so rectangles [4, 2] and [6, 3] are nearly similar, but rectangles [2, 4] and [6, 3] are not. Given an array of rectangles with the lengths of their sides, calculate the number of pairs of nearly similar rectangles in the array.

For example, let's say there are *n = 4* rectangles, and *sides = [[5, 10], [10, 10], [3, 6], [9, 9]]*. In this case, the first and third rectangles, with sides [5, 10] and [3, 6], are nearly similar because 5/3 = 10/6. Also, the second and fourth rectangles, with sides [10, 10] and [9, 9], are nearly similar because 10/9 = 10/9. This means there are 2 pairs of nearly similar rectangles in the array. Therefore, the answer is 2.

Function Description
Complete the function *nearlySimilarRectangles* in the editor below.

nearlySimilarRectangles has the following parameter:
int *sides[n][2]*: a 2-dimensional integer array where the i^{th} row denotes the sides of the i^{th} rectangle
Returns:
int: the number of nearly similar rectangles in the array

Constraints

- $1 \le n \le 10^5$
- $1 \le sides[i][0], sides[i][1] \le 10^{15}$

▸ **Input Format For Custom Testing**

▾ **Sample Case 0**

Sample Input For Custom Testing

```
3
2
4 8
15 30
25 50
```

Sample Output

```
3
```

Explanation
In this example, *n = 3* and *sides = [[4, 8], [15, 30], [25, 50]]*.

- The first and second rectangles, with sides [4, 8] and [15, 30], are nearly similar because 4/15 = 8/30.
- The first and third rectangles, with sides [4, 8] and [25, 50], are nearly similar because 4/25 = 8/50.
- The second and third rectangles, with sides [15, 30] and [25, 50] are nearly similar because 15/25 = 30/50.

This means there are 3 pairs of nearly similar rectangles in this array. Therefore, the answer is 3.

▾ **Sample Case 1**

Sample Input For Custom Testing

```
5
2
2 1
10 7
9 6
6 9
7 3
```

Sample Output

```
0
```

Explanation
In this example, n = 5 and *sides = [[2, 1], [10, 7], [9, 5], [6, 9], [7,3]]*. There are no pairs of nearly similar rectangles in this array. Therefore, the answer is 0.

Language Python 3 | Environment

Autocomplete Ready

```
ratio
    for ratio, count in ratio_count.items():
        if count > 1:
            similar_pairs += count * (count - 1) // 2
        # Debug statement
        print(f"Ratio: {ratio}, Count: {count}, Similar pairs so far: {similar_pairs}")

    # Return the total number of similar pairs
    return similar_pairs
```

Line: 30 Col: 1

Test Results | Custom Input | Run Code | Run Tests | Submit

Compiled successfully. Run all test cases

```
25 50
```

Your Output (stdout)

```
~ no response on stdout ~
```

Expected Output

```
3
```