2001. Number of Pairs of Interchangeable Rectangles

Medium | Topics | Companies | Hint

You are given `n` rectangles represented by a **0-indexed** 2D integer array `rectangles`, where `rectangles[i] =` $[width_i, height_i]$ denotes the width and height of the i^{th} rectangle.

Two rectangles `i` and `j` (`i < j`) are considered **interchangeable** if they have the **same** width-to-height ratio. More formally, two rectangles are **interchangeable** if $width_i/height_i == width_j/height_j$ (using decimal division, not integer division).

Return *the **number** of pairs of **interchangeable** rectangles in* `rectangles`.

Example 1:

```
Input: rectangles = [[4,8],[3,6],[10,20],[15,30]]
Output: 6
Explanation: The following are the interchangeable pairs of rectangles by index (0-indexed):
- Rectangle 0 with rectangle 1: 4/8 == 3/6.
- Rectangle 0 with rectangle 2: 4/8 == 10/20.
- Rectangle 0 with rectangle 3: 4/8 == 15/30.
- Rectangle 1 with rectangle 2: 3/6 == 10/20.
- Rectangle 1 with rectangle 3: 3/6 == 15/30.
- Rectangle 2 with rectangle 3: 10/20 == 15/30.
```

Example 2:

```
Input: rectangles = [[4,5],[7,8]]
Output: 0
Explanation: There are no interchangeable pairs of rectangles.
```

Constraints:

- `n == rectangles.length`
- $1 <= n <= 10^5$
- `rectangles[i].length == 2`
- $1 <= width_i, height_i <= 10^5$

Approach 1: Using sort ( )

→ Store width / height ratio of each

→ Store width/height ratio of each rectangle in a ratios vector.

→ now sort ratios vector.

→ now use two pointers approach to count.

[0.5 0.5 0.5 0.6 0.6 0.7]

```
l=0 , r=0
while( r < ratios.size() )
{
    if( ratios[l] == ratios[r] )
    {
        ans = ans + r - l
        r++
    }
    else
    {
        l = r
        r++
    }
}
```

$T(n): O(n) + O(n\log n) + O(n)$

$S(n): O(n)$

Approach 2: Using hashmap

→ Store width/height of each rectangle in a hashmap with width/height as key and frequency of width/height as value.

→ now traverse map and count ans using frequencies.

```
for ( auto i : map)
{
    x = i.second
    ans = ans + xC2
}
```

$T(n) : O(n) + O(n)$

$S(n) : O(n)$