You are given an array rectangles where rectangles[i] = [li, wi] represents the ith rectangle of length li and width wi.

You can cut the ith rectangle to form a square with a side length of k if both k <= li and k <= wi. For example, if you have a rectangle [4,6], you can cut it to get a square with a side length of at most 4.

Let maxLen be the side length of the **largest** square you can obtain from any of the given rectangles.

Return *the* ***number*** *of rectangles that can make a square with a side length of* maxLen.

**Example 1:**

Input: rectangles = [[5,8],[3,9],[5,12],[16,5]]  
Output: 3  
Explanation: The largest squares you can get from each rectangle are of lengths [5,3,5,5].  
The largest possible square is of length 5, and you can get it out of 3 rectangles.

**Example 2:**

Input: rectangles = [[2,3],[3,7],[4,3],[3,7]]  
Output: 3

**Constraints:**

* 1 <= rectangles.length <= 1000
* rectangles[i].length == 2
* 1 <= li, wi <= 109
* li != wi