

1274. Number of Ships in a Rectangle

Hard👍 223👤 36💖 Add to List🔖 Share

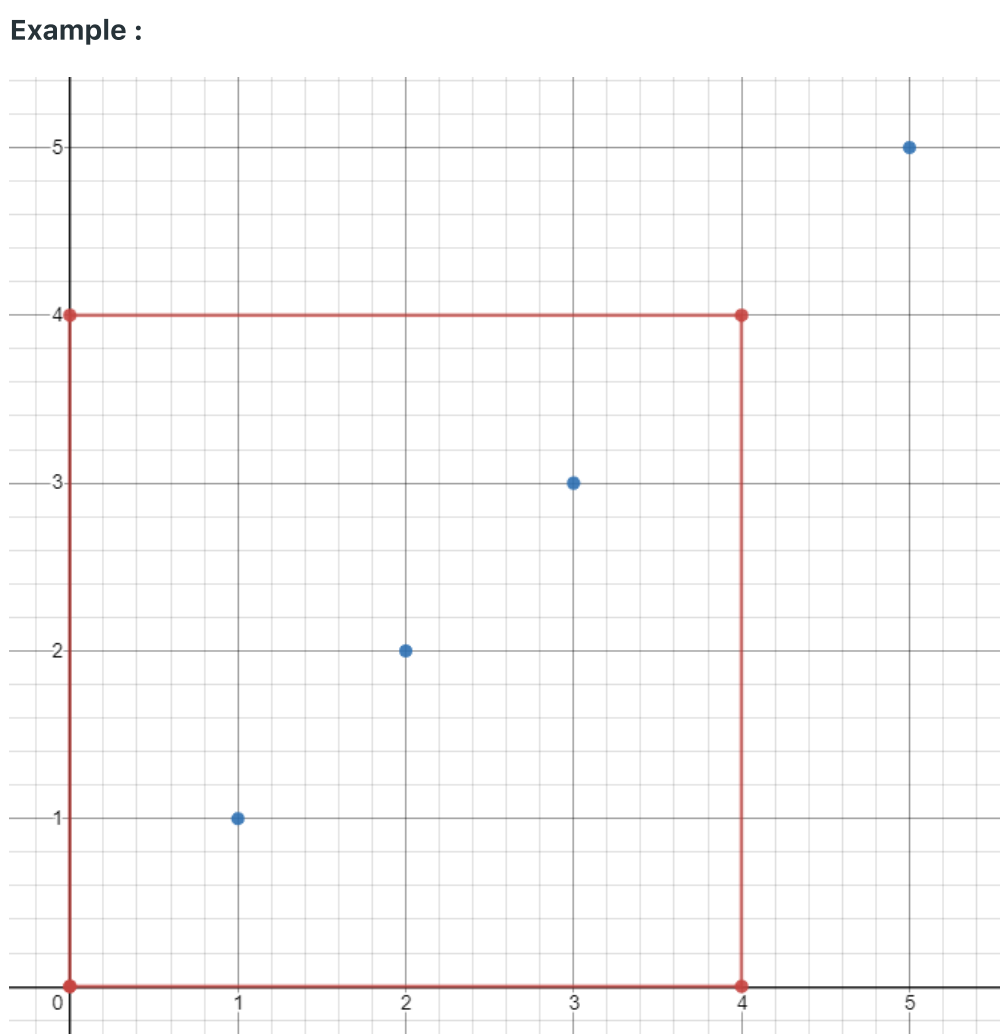
(This problem is an *interactive problem*.)

Each ship is located at an integer point on the sea represented by a cartesian plane, and each integer point may contain at most 1 ship.

You have a function `Sea.hasShips(topRight, bottomLeft)` which takes two points as arguments and returns `true` if there is at least one ship in the rectangle represented by the two points, including on the boundary.

Given two points: the top right and bottom left corners of a rectangle, return the number of ships present in that rectangle. It is guaranteed that there are **at most 10 ships** in that rectangle.

Submissions making **more than 400 calls** to `hasShips` will be judged *Wrong Answer*. Also, any solutions that attempt to circumvent the judge will result in disqualification.



**Input:**  
ships = [[1,1],[2,2],[3,3],[5,5]], topRight = [4,4], bottomLeft = [0,0]  
**Output:** 3  
**Explanation:** From [0,0] to [4,4] we can count 3 ships within the range.

**Example 2:**

**Input:** ans = [[1,1],[2,2],[3,3]], topRight = [1000,1000], bottomLeft = [0,0]  
**Output:** 3

- Constraints:**
- On the input `ships` is only given to initialize the map internally. You must solve this problem "blindfolded". In other words, you must find the answer using the given `hasShips` API, without knowing the `ships` position.
  - $0 \leq \text{bottomLeft}[0] \leq \text{topRight}[0] \leq 1000$
  - $0 \leq \text{bottomLeft}[1] \leq \text{topRight}[1] \leq 1000$
  - $\text{topRight} \neq \text{bottomLeft}$

Accepted 12,215 Submissions 18,487

Seen this question in a real interview before? 

Yes

No

Companies 🍷 ⓘ ^

0 ~ 6 months6 months ~ 1 year1 year ~ 2 years

Bloomberg5

Related Topics ^

ArrayDivide and ConquerInteractive

Hide Hint 1 ^

Use divide and conquer technique.

Hide Hint 2 ^

Divide the query rectangle into 4 rectangles.

Hide Hint 3 ^

Use recursion to continue with the rectangles that has ships only.

f JavaAutocomplete

```
1 // /**
2  * // This is Sea's API interface.
3  * // You should not implement it, or speculate about its implementation
4  * class Sea {
5  *     public boolean hasShips(int[] topRight, int[] bottomLeft);
6  * }
7  */
8
9 class Solution {
10     public int countShips(Sea sea, int[] topRight, int[] bottomLeft) {
11
12     }
13 }
```