Problem 3189: Minimum Moves to Get a Peaceful Board

Problem Information

Difficulty: Medium
Acceptance Rate: 76.03%
Paid Only: Yes
Tags: Array, Greedy, Sorting, Counting Sort

Problem Description

Given a 2D array `rooks` of length `n`, where `rooks[i] = [xi, yi]` indicates the position of a rook on an `n x n` chess board. Your task is to move the rooks **1 cell** at a time vertically or horizontally (to an _adjacent_ cell) such that the board becomes **peaceful**.

A board is **peaceful** if there is **exactly** one rook in each row and each column.

Return the **minimum** number of moves required to get a _peaceful board_.

Note that **at no point** can there be two rooks in the same cell.

Example 1:

Input: rooks = [[0,0],[1,0],[1,1]]

Output: 3

Explanation:

![](https://assets.leetcode.com/uploads/2024/06/17/ex1-edited.gif)

Example 2:

Input: rooks = [[0,0],[0,1],[0,2],[0,3]]

Output: 6

Explanation:

![](https://assets.leetcode.com/uploads/2024/06/17/ex2-edited.gif)

Constraints:

* `1 <= n == rooks.length <= 500` * `0 <= xi, yi <= n - 1` * The input is generated such that there are no 2 rooks in the same cell.

Code Snippets

C++:

```
class Solution {
public:
int minMoves(vector<vector<int>>& rooks) {

}
};
```

Java:

```
class Solution {
public int minMoves(int[][] rooks) {

}
}
```

Python3:

```
class Solution:
def minMoves(self, rooks: List[List[int]]) -> int:
```