

Problem 757: Set Intersection Size At Least Two

Problem Information

Difficulty: Hard

Acceptance Rate: 57.16%

Paid Only: No

Tags: Array, Greedy, Sorting

Problem Description

You are given a 2D integer array `intervals` where `intervals[i] = [starti, endi]` represents all the integers from `starti` to `endi` inclusively.

A **containing set** is an array `nums` where each interval from `intervals` has **at least two** integers in `nums` .

* For example, if `intervals = [[1,3], [3,7], [8,9]]` , then `[1,2,4,7,8,9]` and `[2,3,4,8,9]` are **containing sets**.

Return _the minimum possible size of a containing set_.

Example 1:

Input: intervals = [[1,3],[3,7],[8,9]] **Output:** 5 **Explanation:** let nums = [2, 3, 4, 8, 9]. It can be shown that there cannot be any containing array of size 4.

Example 2:

Input: intervals = [[1,3],[1,4],[2,5],[3,5]] **Output:** 3 **Explanation:** let nums = [2, 3, 4]. It can be shown that there cannot be any containing array of size 2.

Example 3:

Input: intervals = [[1,2],[2,3],[2,4],[4,5]] **Output:** 5 **Explanation:** let nums = [1, 2, 3, 4, 5]. It can be shown that there cannot be any containing array of size 4.

****Constraints:****

* `1 <= intervals.length <= 3000` * `intervals[i].length == 2` * `0 <= starti < endi <= 108`

Code Snippets

C++:

```
class Solution {  
public:  
    int intersectionSizeTwo(vector<vector<int>>& intervals) {  
  
    }  
};
```

Java:

```
class Solution {  
public int intersectionSizeTwo(int[][] intervals) {  
  
}  
}
```

Python3:

```
class Solution:  
    def intersectionSizeTwo(self, intervals: List[List[int]]) -> int:
```