Problem 3323: Minimize Connected Groups by Inserting Interval

Problem Information

Difficulty: Medium
Acceptance Rate: 50.49%
Paid Only: Yes
Tags: Array, Binary Search, Sliding Window, Sorting

Problem Description

You are given a 2D array `intervals`, where `intervals[i] = [starti, endi]` represents the start and the end of interval `i`. You are also given an integer `k`.

You must add **exactly one** new interval `[startnew, endnew]` to the array such that:

* The length of the new interval, `endnew - startnew`, is at most `k`. * After adding, the number of **connected groups** in `intervals` is **minimized**.

A **connected group** of intervals is a maximal collection of intervals that, when considered together, cover a continuous range from the smallest point to the largest point with no gaps between them. Here are some examples:

* A group of intervals `[[1, 2], [2, 5], [3, 3]]` is connected because together they cover the range from 1 to 5 without any gaps. * However, a group of intervals `[[1, 2], [3, 4]]` is not connected because the segment `(2, 3)` is not covered.

Return the **minimum** number of connected groups after adding **exactly one** new interval to the array.

Example 1:

Input: intervals = [[1,3],[5,6],[8,10]], k = 3

Output: 2

Explanation:

After adding the interval `[3, 5]`, we have two connected groups: `[[1, 3], [3, 5], [5, 6]]` and `[[8, 10]]`.

Example 2:

Input: intervals = [[5,10],[1,1],[3,3]], k = 1

Output: 3

Explanation:

After adding the interval `[1, 1]`, we have three connected groups: `[[1, 1], [1, 1]]`, `[[3, 3]]`, and `[[5, 10]]`.

Constraints:

* `1 <= intervals.length <= 105` * `intervals[i] == [starti, endi]` * `1 <= starti <= endi <= 109` * `1 <= k <= 109`

Code Snippets

C++:

```
class Solution {
public:
int minConnectedGroups(vector<vector<int>>& intervals, int k) {

}
};
```

Java:

```
class Solution {
public int minConnectedGroups(int[][] intervals, int k) {

}
}
```

Python3:

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