Problem 986: Interval List Intersections

Problem Information

Difficulty: Medium
Acceptance Rate: 72.84%
Paid Only: No
Tags: Array, Two Pointers, Line Sweep

Problem Description

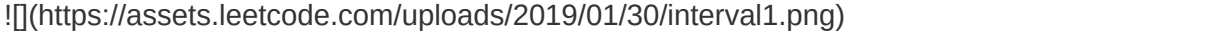
You are given two lists of closed intervals, `firstList` and `secondList`, where `firstList[i] = [starti, endi]` and `secondList[j] = [startj, endj]`. Each list of intervals is pairwise **disjoint** and in **sorted order**.

Return _the intersection of these two interval lists_.

A **closed interval** `[a, b]` (with `a <= b`) denotes the set of real numbers `x` with `a <= x <= b`.

The **intersection** of two closed intervals is a set of real numbers that are either empty or represented as a closed interval. For example, the intersection of `[1, 3]` and `[2, 4]` is `[2, 3]`.

Example 1:



Input: firstList = [[0,2],[5,10],[13,23],[24,25]], secondList = [[1,5],[8,12],[15,24],[25,26]] **Output:** [[1,2],[5,5],[8,10],[15,23],[24,24],[25,25]]

Example 2:

Input: firstList = [[1,3],[5,9]], secondList = [] **Output:** []

Constraints:

* `0 <= firstList.length, secondList.length <= 1000` * `firstList.length + secondList.length >= 1` * `0 <= starti < endi <= 109` * `endi < starti+1` * `0 <= startj < endj <= 109 ` * `endj < startj+1`

Code Snippets

C++:

```
class Solution {
public:
vector<vector<int>> intervalIntersection(vector<vector<int>>& firstList,
vector<vector<int>>& secondList) {

}
};
```

Java:

```
class Solution {
public int[][] intervalIntersection(int[][] firstList, int[][] secondList) {

}
}
```

Python3:

```
class Solution:
def intervalIntersection(self, firstList: List[List[int]], secondList:
List[List[int]]) -> List[List[int]]:
```