986. Interval List Intersections

Given two lists of closed intervals, each list of intervals is pairwise disjoint and in sorted order.

Return the intersection of these two interval lists.

(Formally, 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 is either empty, or can be represented as a closed interval. For example, the intersection of [1, 3] and [2, 4] is [2, 3].)

Example 1:

```
Input: A = [[0,2],[5,10],[13,23],[24,25]], B = [[1,5],[8,12],[15,24],[25,26]]
Output: [[1,2],[5,5],[8,10],[15,23],[24,24],[25,25]]
Reminder: The inputs and the desired output are lists of Interval objects, and not ar
rays or lists.
```

```
class Solution {
    public int[][] intervalIntersection(int[][] A, int[][] B) {

        // two pointers
        if (A == null || A.length == 0 || B == null || B.length == 0)
            return new int[][] {};

        List<int[]> res = new ArrayList<>();

        int i = 0, j = 0;
        int startMax, endMin;

        while(i < A.length && j < B.length) {
            startMax = Math.max(A[i][0], B[j][0]);
            endMin = Math.min(A[i][1], B[j][1]);

            if (endMin >= startMax) {
                res.add(new int[] {startMax, endMin});
            }

            if (A[i][1] == endMin)
                i++;

            if (B[j][1] == endMin)
                j++;
        }

        return res.toArray(new int[0][0]);
    }
}
```