Problem 1213: Intersection of Three Sorted Arrays

Problem Information

Difficulty: Easy
Acceptance Rate: 80.04%
Paid Only: Yes
Tags: Array, Hash Table, Binary Search, Counting

Problem Description

Given three integer arrays `arr1`, `arr2` and `arr3` **sorted** in **strictly increasing** order, return a sorted array of **only** the integers that appeared in **all** three arrays.

Example 1:

Input: arr1 = [1,2,3,4,5], arr2 = [1,2,5,7,9], arr3 = [1,3,4,5,8] **Output:** [1,5]
Explanation: Only 1 and 5 appeared in the three arrays.

Example 2:

Input: arr1 = [197,418,523,876,1356], arr2 = [501,880,1593,1710,1870], arr3 = [521,682,1337,1395,1764] **Output:** []

Constraints:

* `1 <= arr1.length, arr2.length, arr3.length <= 1000` * `1 <= arr1[i], arr2[i], arr3[i] <= 2000`

Code Snippets

C++:

```
class Solution {
public:
vector<int> arraysIntersection(vector<int>& arr1, vector<int>& arr2,
```

```
    vector<int>& arr3) {


    }
};
```

Java:

```
class Solution {
public List<Integer> arraysIntersection(int[] arr1, int[] arr2, int[] arr3) {


    }
}
```

Python3:

```
class Solution:
    def arraysIntersection(self, arr1: List[int], arr2: List[int], arr3:
    List[int]) -> List[int]:
```