# 1213. Intersection of Three Sorted Arrays

Easy | 374 | 21 | Add to List | Share

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`

Accepted 56,642 | Submissions 70,943

Seen this question in a real interview before? Yes No

Companies

0 ~ 6 months | 6 months ~ 1 year | 1 year ~ 2 years

Facebook | 3

Related Topics

Array | Hash Table | Binary Search | Counting

Similar Questions

Intersection of Two Arrays — Easy

Hide Hint 1

Count the frequency of all elements in the three arrays.

Hide Hint 2

The elements that appeared in all the arrays would have a frequency of 3.

Java | Autocomplete

```java
class Solution {
    public List<Integer> arraysIntersection(int[] arr1, int[] arr2, int[] arr3) {
        List<Integer> ans = new ArrayList <>();
        // prepare three pointers to iterate through three arrays
        // p1, p2, and p3 point to the beginning of arr1, arr2, and arr3 accordingly
        int p1 = 0, p2 = 0, p3 = 0;

        while (p1 < arr1.length && p2 < arr2.length && p3 < arr3.length) {

            if (arr1[p1] == arr2[p2] && arr2[p2] == arr3[p3]) {
                ans.add(arr1[p1]);
                p1++;
                p2++;
                p3++;
            } else {
                if (arr1[p1] < arr2[p2]) {
                    p1++;
                } else if (arr2[p2] < arr3[p3]) {
                    p2++;
                } else {
                    p3++;
                }

            }
        }
        return ans;
    }
}
```