Problem List | Run | Submit

Description | Editorial | Solutions | Submissions

1213. Intersection of Three Sorted Arrays

Premium

Solved

Easy | Topics | Companies | Hint

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`

Seen this question in a real interview before? 1/5

Yes | No

Accepted 85.9K | Submissions 107.7K | Acceptance Rate 79.8%

Topics

Companies

Hint 1

Hint 2

Similar Questions

Discussion (2)

580 | 2

Code

Python3 • Auto

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

        dictx = {}

        for each in arr1:
            if each not in dictx:
                dictx[each] = 1
            else:
                dictx[each] += 1

        for each in arr2:
            if each not in dictx:
                dictx[each] = 1
            else:
                dictx[each] += 1

        for each in arr3:
            if each not in dictx:
                dictx[each] = 1
            else:
                dictx[each] += 1

        ans = []
```

Saved

Ln 9, Col 18

Testcase | Test Result

Accepted Runtime: 60 ms

• Case 1 • Case 2

Input

arr1 =

[1,2,3,4,5]

arr2 =

[1,2,5,7,9]