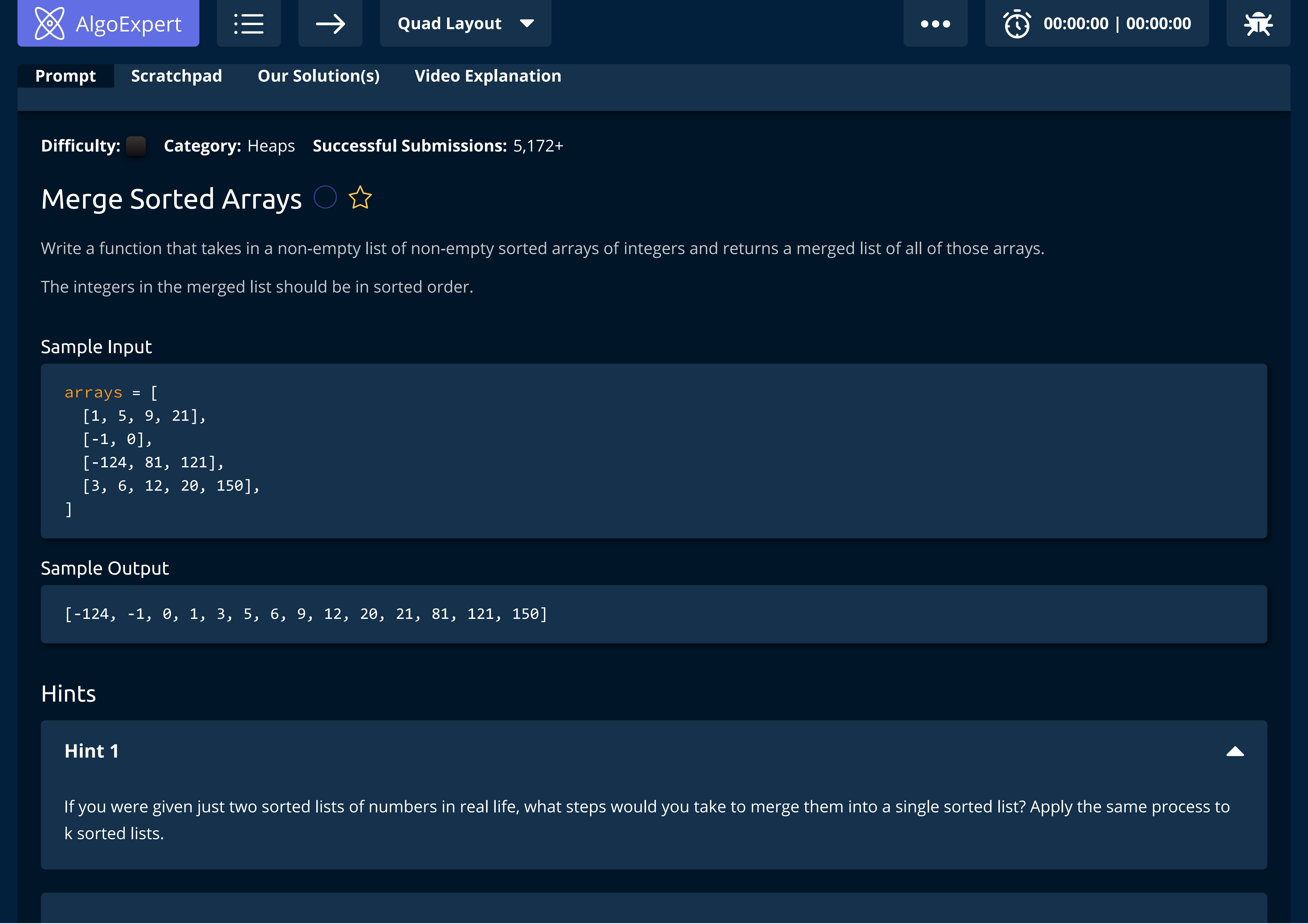


Difficulty: ■ **Category:** Heaps **Successful Submissions:** 5,172+

Merge Sorted Arrays ○ ☆

Write a function that takes in a non-empty list of non-empty sorted arrays of integers and returns a merged list of all of those arrays.

The integers in the merged list should be in sorted order.

Sample Input

```
arrays = [
  [1, 5, 9, 21],
  [-1, 0],
  [-124, 81, 121],
  [3, 6, 12, 20, 150],
]
```

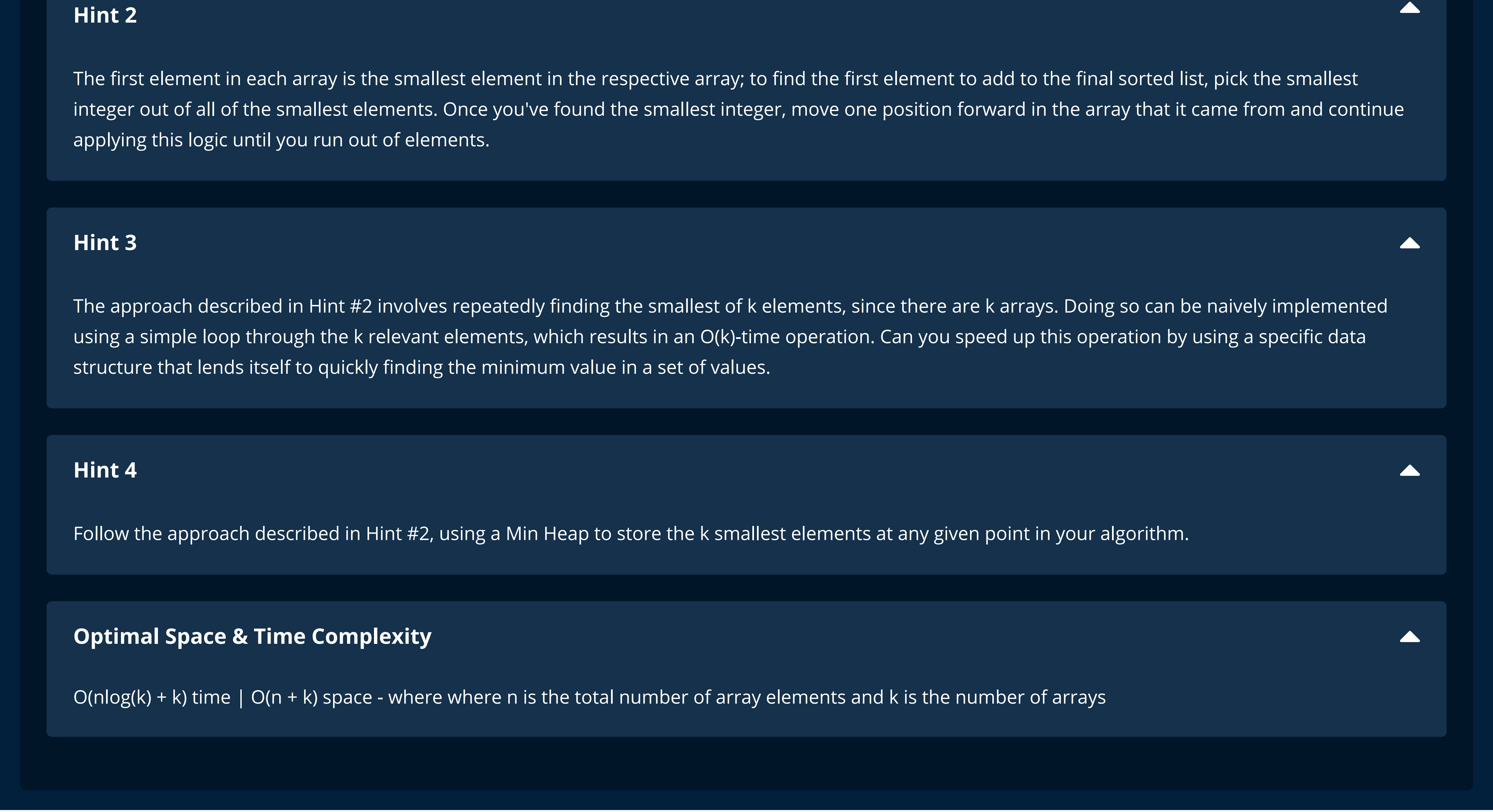
Sample Output

```
[-124, -1, 0, 1, 3, 5, 6, 9, 12, 20, 21, 81, 121, 150]
```

Hints

Hint 1

If you were given just two sorted lists of numbers in real life, what steps would you take to merge them into a single sorted list? Apply the same process to k sorted lists.



Hint 2

The first element in each array is the smallest element in the respective array; to find the first element to add to the final sorted list, pick the smallest integer out of all of the smallest elements. Once you've found the smallest integer, move one position forward in the array that it came from and continue applying this logic until you run out of elements.

Hint 3

The approach described in Hint #2 involves repeatedly finding the smallest of k elements, since there are k arrays. Doing so can be naively implemented using a simple loop through the k relevant elements, which results in an O(k)-time operation. Can you speed up this operation by using a specific data structure that lends itself to quickly finding the minimum value in a set of values.

Hint 4

Follow the approach described in Hint #2, using a Min Heap to store the k smallest elements at any given point in your algorithm.

Optimal Space & Time Complexity

O(nlog(k) + k) time | O(n + k) space - where where n is the total number of array elements and k is the number of arrays