Our Solution(s) Run Code

Solution 1

```go
// Copyright © 2020 AlgoExpert, LLC. All rights reserved.

package main

import (
  "math"
  "sort"
)

// O(nlog(n) + mlog(m)) time | O(1) space
func SmallestDifference(array1, array2 []int) []int {
  sort.Ints(array1)
  sort.Ints(array2)
  idxOne, idxTwo := 0, 0
  smallest, current := math.MaxInt32, math.MaxInt32
  smallestPair := []int{}
  for idxOne < len(array1) && idxTwo < len(array2) {
    first, second := array1[idxOne], array2[idxTwo]
    if first < second {
      current = second - first
      idxOne += 1
    } else if second < first {
      current = first - second
      idxTwo += 1
    } else {
      return []int{first, second}
    }
    if smallest > current {
      smallest = current
      smallestPair = []int{first, second}
    }
  }
  return smallestPair
}
```

Your Solutions Run Code

Solution 1 Solution 2 Solution 3

```go
package main

func SmallestDifference(array1, array2 []int) []int {
  // Write your code here.
  return nil
}

```

Custom Output Raw Output Submit Code

Run or submit code when you're ready.