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

class Program {
  struct Item {
    var arrayIdx: Int
    var num: Int
  }

  // O(nk) time | O(n + k) space - where where n is the total
  // number of array elements and k is the number of arrays
  static func mergeSortedArrays(_ arrays: [[Int]]) -> [Int] {
    var sortedList = [Int]()
    var elementIdxs = [Int]()
    for el in arrays {
      elementIdxs.append(0)
    }

    while true {
      var smallestItems = [Item]()
      for arrayIdx in 0 ..< arrays.count {
        var relevantArray = arrays[arrayIdx]
        var elementIdx = elementIdxs[arrayIdx]
        if elementIdx == relevantArray.count {
          continue
        }
        smallestItems.append(Item(arrayIdx: arrayIdx, num: relevantArray[elementIdx]))
      }

      if smallestItems.count == 0 {
        break
      }
      var nextItem = getMinValue(&smallestItems)
      sortedList.append(nextItem.num)
      elementIdxs[nextItem.arrayIdx] += 1
    }
    return sortedList
  }

  static func getMinValue(_ items: inout [Item]) -> Item {
    var minValueItem = items[0]
    for i in 1 ..< items.count {
      if items[i].num < minValueItem.num {
        minValueItem = items[i]
      }
    }
    return minValueItem
  }
}

```