```
1  // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3  using System;
4  using System.Linq;
5  using System.Collections.Generic;
6
7  public class Program {
8    // O(nk) time | O(n + k) space - where where n is the total
9    // number of array elements and k is the number of arrays
10   public static List<int> MergeSortedArrays(List<List<int> > arrays) {
11     List<int> sortedList = new List<int>();
12     List<int> elementIdxs = Enumerable.Repeat(0, arrays.Count).ToList();
13     while (true) {
14       List<Item> smallestItems = new List<Item>();
15       for (int arrayIdx = 0; arrayIdx < arrays.Count; arrayIdx++) {
16         List<int> relevantArray = arrays[arrayIdx];
17         int elementIdx= elementIdxs[arrayIdx];
18         if (elementIdx== relevantArray.Count) continue;
19         smallestItems.Add(new Item(arrayIdx, relevantArray[elementIdx]));
20       }
21       if (smallestItems.Count == 0) break;
22       Item nextItem = getMinValue(smallestItems);
23       sortedList.Add(nextItem.num);
24       elementIdxs[nextItem.arrayIdx] =  elementIdxs[nextItem.arrayIdx] + 1;
25     }
26
27     return sortedList;
28   }
29
30   public static Item getMinValue(List<Item> items) {
31     int minValueIdx = 0;
32     for (int i = 1; i < items.Count; i++) {
33       if (items[i].num < items[minValueIdx].num) minValueIdx = i;
34     }
35     return items[minValueIdx];
36   }
37
38   public class Item {
39     public int arrayIdx;
40     public int num;
41
42     public Item(int arrayIdx, int num) {
43       this.arrayIdx = arrayIdx;
44       this.num = num;
45     }
46   }
47 }
48
```