AlgoExpert Quad Layout C# 12px Sublime Monokai 00:00:00

Prompt Scratchpad Our Solution(s) Video Explanation Run Code

Solution 1 Solution 2

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


public class Program {
  // Best: O(nlog(n)) time | O(n) space
  // Average: O(nlog(n)) time | O(n) space
  // Worst: O(nlog(n)) time | O(n) space
  public static int[] MergeSort(int[] array) {
    if (array.Length <= 1) {
      return array;
    }
    int[] auxiliaryArray = (int[]) array.Clone();
    MergeSort(array, 0, array.Length - 1, auxiliaryArray);
    return array;
  }

  public static void MergeSort(int[] mainArray, int startIdx, int endIdx,
    int[] auxiliaryArray) {
    if (startIdx == endIdx) {
      return;
    }
    int middleIdx = (startIdx + endIdx) / 2;
    MergeSort(auxiliaryArray, startIdx, middleIdx, mainArray);
    MergeSort(auxiliaryArray, middleIdx + 1, endIdx, mainArray);
    doMerge(mainArray, startIdx, middleIdx, endIdx, auxiliaryArray);
  }

  public static void doMerge(int[] mainArray, int startIdx, int middleIdx, int endIdx,
    int[] auxiliaryArray) {
    int k = startIdx;
    int i = startIdx;
    int j = middleIdx + 1;
    while (i <= middleIdx && j <= endIdx) {
      if (auxiliaryArray[i] <= auxiliaryArray[j]) {
        mainArray[k++] = auxiliaryArray[i++];
      } else {
        mainArray[k++] = auxiliaryArray[j++];
      }
    }
    while (i <= middleIdx) {
      mainArray[k++] = auxiliaryArray[i++];
    }
    while (j <= endIdx) {
      mainArray[k++] = auxiliaryArray[j++];
    }
  }
}

```