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

#include <vector>
using namespace std;

vector<int> mergeSortedArrays(vector<int> leftHalf, vector<int> rightHalf);

// Best: O(nlog(n)) time | O(nlog(n)) space
// Average: O(nlog(n)) time | O(nlog(n)) space
// Worst: O(nlog(n)) time | O(nlog(n)) space
vector<int> mergeSort(vector<int> array) {
  if (array.size() <= 1) {
    return array;
  }
  int middleIdx = array.size() / 2;
  vector<int> leftHalf(array.begin(), array.begin() + middleIdx);
  vector<int> rightHalf(array.begin() + middleIdx, array.end());
  return mergeSortedArrays(mergeSort(leftHalf), mergeSort(rightHalf));
}

vector<int> mergeSortedArrays(vector<int> leftHalf, vector<int> rightHalf) {
  vector<int> sortedArray(leftHalf.size() + rightHalf.size(), 0);
  int k = 0;
  int i = 0;
  int j = 0;
  while (i < leftHalf.size() && j < rightHalf.size()) {
    if (leftHalf[i] <= rightHalf[j]) {
      sortedArray[k++] = leftHalf[i++];
    } else {
      sortedArray[k++] = rightHalf[j++];
    }
  }
  while (i < leftHalf.size()) {
    sortedArray[k++] = leftHalf[i++];
  }
  while (j < rightHalf.size()) {
    sortedArray[k++] = rightHalf[j++];
  }
  return sortedArray;
}

```