Our Solution(s) Run Code

Solution 1 Solution 2

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

#include <vector>
using namespace std;

void mergeSortHelper(vector<int> *mainArray, int startIdx, int endIdx
                     vector<int> *auxiliaryArray);
void doMerge(vector<int> *mainArray, int startIdx, int middleIdx, int
             vector<int> *auxiliaryArray);

// Best: O(nlog(n)) time | O(n) space
// Average: O(nlog(n)) time | O(n) space
// Worst: O(nlog(n)) time | O(n) space
vector<int> mergeSort(vector<int> array) {
  if (array.size() <= 1) {
    return array;
  }
  vector<int> auxiliaryArray = array;
  mergeSortHelper(&array, 0, array.size() - 1, &auxiliaryArray);
  return array;
}

void mergeSortHelper(vector<int> *mainArray, int startIdx, int endIdx,
                     vector<int> *auxiliaryArray) {
  if (startIdx == endIdx) {
    return;
  }
  int middleIdx = (startIdx + endIdx) / 2;
  mergeSortHelper(auxiliaryArray, startIdx, middleIdx, mainArray);
  mergeSortHelper(auxiliaryArray, middleIdx + 1, endIdx, mainArray);
  doMerge(mainArray, startIdx, middleIdx, endIdx, auxiliaryArray);
}

```

Your Solutions Run Code

Solution 1 Solution 2 Solution 3

```cpp
#include <vector>
using namespace std;

vector<int> mergeSort(vector<int> array) {
  // Write your code here.
  return {};
}

```

Our Tests

Custom Output Submit Code

Run or submit code when you're ready.