

Our Solution(s)

Run Code

Your Solutions

Run Code

Solution 1

Solution 2

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 #include <vector>
4 using namespace std;
5
6 void mergeSortHelper(vector<int> *mainArray, int startIdx, int endIdx,
7                     vector<int> *auxiliaryArray);
8 void doMerge(vector<int> *mainArray, int startIdx, int middleIdx, int
9             vector<int> *auxiliaryArray);
10
11 // Best: O(nlog(n)) time | O(n) space
12 // Average: O(nlog(n)) time | O(n) space
13 // Worst: O(nlog(n)) time | O(n) space
14 vector<int> mergeSort(vector<int> array) {
15     if (array.size() <= 1) {
16         return array;
17     }
18     vector<int> auxiliaryArray = array;
19     mergeSortHelper(&array, 0, array.size() - 1, &auxiliaryArray);
20     return array;
21 }
22
23 void mergeSortHelper(vector<int> *mainArray, int startIdx, int endIdx,
24                     vector<int> *auxiliaryArray) {
25     if (startIdx == endIdx) {
26         return;
27     }
28     int middleIdx = (startIdx + endIdx) / 2;
29     mergeSortHelper(auxiliaryArray, startIdx, middleIdx, mainArray);
30     mergeSortHelper(auxiliaryArray, middleIdx + 1, endIdx, mainArray);
31     doMerge(mainArray, startIdx, middleIdx, endIdx, auxiliaryArray);
32 }
33
```

Solution 1

Solution 2

Solution 3

```
1 #include <vector>
2 using namespace std;
3
4 vector<int> mergeSort(vector<int> array) {
5     // Write your code here.
6     return {};
7 }
8
```

Our Tests

Custom Output

Submit Code

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

57

58

59

60

61

62

63

64

65

66

67

68

69

70

71

72

73

74

75

76

77

78

79

80

81

82

83

84

85

86

87

88

89

90

91

92

93

94

95

96

97

98

99

100

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

57

58

59

60

61

62

63

64

65

66

67

68

69

70

71

72

73

74

75

76

77

78

79

80

81

82

83

84

85

86

87

88

89

90

91

92

93

94

95

96

97

98

99

100

```
10
11
12 def test_case_27, 100, 2
13   assert_equal expected_27, 50
14   assert_equal sum_27, 50 + expected_27
15 end
16
17 def test_case_28, 100, 2
18   assert_equal expected_28, 50
19   assert_equal sum_28, 50 + expected_28
20 end
21
22 def test_case_29, 100, 2
23   assert_equal expected_29, 5, 50
24   assert_equal sum_29, 5, 50 + expected_29
25 end
```

Run or submit code when you're ready.