AlgoExpert   Quad Layout   00:00:00 | 00:00:00

Prompt   **Scratchpad**   **Our Solution(s)**   **Video Explanation**

Difficulty: ⬛   **Category:** Sorting   **Successful Submissions:** 9,707+

Merge Sort ○ ☆

Write a function that takes in an array of integers and returns a sorted version of that array. Use the Merge Sort algorithm to sort the array.

If you're unfamiliar with Merge Sort, we recommend watching the Conceptual Overview section of this question's video explanation before starting to code.

Sample Input

```
array = [8, 5, 2, 9, 5, 6, 3]
```

Sample Output

```
[2, 3, 5, 5, 6, 8, 9]
```

Hints

Hint 1

Merge Sort works by cutting an array in two halves, respectively sorting those two halves by performing some special logic, and then merging the two newly-sorted halves into one sorted array. The respective sorting of the two halves is done by reapplying the Merge Sort algorithm / logic on each half until single-element halves are obtained; these single-element arrays are sorted by nature and can very easily be merged back together.

Hint 2

Divide the input array in two halves by finding the middle-most index in the array and slicing the two halves around that index. Then, recursively apply Merge Sort to each half, and finally merge them into one single, sorted array by iterating through their values and progressively adding them to the new array in ascending order.

Hint 3

Your implementation of Merge Sort almost certainly uses a lot of auxiliary space and likely does not sort the input array in place. What is the space complexity of your algorithm? Can you implement a version of the algorithm using only one additional array of the same length as the input array, and can this version sort the input array in place?

Optimal Space & Time Complexity

Best: O(nlog(n)) time | O(n) space - where n is the length of the input array Average: O(nlog(n)) time | O(n) space - where n is the length of the input array
Worst: O(nlog(n)) time | O(n) space - where n is the length of the input array