**Question 1: What is the time complexity of Merge Sort in the best, worst, and average cases?**

A) O(n)  
B) O(n log n)  
C) O(n^2)  
D) O(log n)

**Question 2: Which of the following statements about Merge Sort is true?**

A) Merge Sort is a comparison-based sorting algorithm.  
B) Merge Sort is an in-place sorting algorithm.  
C) Merge Sort can only be used to sort integers.  
D) Merge Sort is slower than Quick Sort in all cases.

**Question 3: How does Merge Sort work to sort an array?**

A) It divides the array into two halves, sorts each half, and merges them together in sorted order.  
B) It iteratively compares adjacent elements and swaps them if they are in the wrong order.  
C) It repeatedly selects the smallest element and places it in the correct position.  
D) It divides the array into multiple smaller arrays and sorts them individually without merging.

**Question 4: What happens during the "merge" step in Merge Sort?**

A) Two sorted halves of the array are combined in ascending order.  
B) Two sorted halves of the array are combined in descending order.  
C) The array is split into two parts based on the pivot element.  
D) The array is iteratively sorted from left to right.

**Question 5: Which of the following is a real-world scenario where Merge Sort is a suitable algorithm?**

A) Sorting the grades of thousands of students in descending order to identify the top performers.  
B) Sorting a small list of items in an e-commerce website shopping cart.  
C) Sorting a small array of numbers in a game that needs to be done instantly.  
D) Searching for a specific value in a sorted list.

**Answer Key**

**Question 1: What is the time complexity of Merge Sort in the best, worst, and average cases?**

A) O(n)  
B) O(n log n)  
C) O(n^2)  
D) O(log n)

**Question 2: Which of the following statements about Merge Sort is true?**

A) Merge Sort is a comparison-based sorting algorithm.  
B) Merge Sort is an in-place sorting algorithm.  
C) Merge Sort can only be used to sort integers.  
D) Merge Sort is slower than Quick Sort in all cases.

**Question 3: How does Merge Sort work to sort an array?**

A) It divides the array into two halves, sorts each half, and merges them together in sorted order.  
B) It iteratively compares adjacent elements and swaps them if they are in the wrong order.  
C) It repeatedly selects the smallest element and places it in the correct position.  
D) It divides the array into multiple smaller arrays and sorts them individually without merging.

**Question 4: What happens during the "merge" step in Merge Sort?**

A) Two sorted halves of the array are combined in ascending order.  
B) Two sorted halves of the array are combined in descending order.  
C) The array is split into two parts based on the pivot element.  
D) The array is iteratively sorted from left to right.

**Question 5: Which of the following is a real-world scenario where Merge Sort is a suitable algorithm?**

A) Sorting the grades of thousands of students in descending order to identify the top performers.  
B) Sorting a small list of items in an e-commerce website shopping cart.  
C) Sorting a small array of numbers in a game that needs to be done instantly.  
D) Searching for a specific value in a sorted list.