

# Rajalakshmi Engineering College

Name: srisanjay RD  
Email: 240801332@rajalakshmi.edu.in  
Roll no: 240801332  
Phone: 8667212915  
Branch: REC  
Department: I ECE AF  
Batch: 2028  
Degree: B.E - ECE

Scan to verify results



## NeoColab\_REC\_CS23231\_DATA STRUCTURES

### REC\_DS using C\_Week 6\_MCQ\_Updated\_1

Attempt : 1  
Total Mark : 20  
Marks Obtained : 20

#### Section 1 : MCQ

1. Merge sort is \_\_\_\_\_.

**Answer**

Comparison-based sorting algorithm

**Status : Correct**

**Marks : 1/1**

2. In a quick sort algorithm, what role does the pivot element play?

**Answer**

It is used to partition the array

**Status : Correct**

**Marks : 1/1**

3. Why is Merge Sort preferred for sorting large datasets compared to Quick Sort?

**Answer**

Merge Sort has better worst-case time complexity

**Status :** Correct

**Marks :** 1/1

4. Which of the following statements is true about the merge sort algorithm?

**Answer**

It requires additional memory for merging

**Status :** Correct

**Marks :** 1/1

5. Which of the following sorting algorithms is based on the divide and conquer method?

**Answer**

Merge Sort

**Status :** Correct

**Marks :** 1/1

6. Which of the following strategies is used to improve the efficiency of Quicksort in practical implementations?

**Answer**

Choosing the pivot randomly or using the median-of-three method

**Status :** Correct

**Marks :** 1/1

7. Which of the following is not true about QuickSort?

**Answer**

It can be implemented as a stable sort

**Status :** Correct

**Marks :** 1/1

8. Let P be a quick sort program to sort numbers in ascending order using the first element as a pivot. Let  $t_1$  and  $t_2$  be the number of comparisons made by P for the inputs  $\{1, 2, 3, 4, 5\}$  and  $\{4, 1, 5, 3, 2\}$ , respectively. Which one of the following holds?

**Answer**

$t_1 > t_2$

**Status :** Correct

**Marks :** 1/1

9. Is Merge Sort a stable sorting algorithm?

**Answer**

Yes, always stable.

**Status :** Correct

**Marks :** 1/1

10. Which of the following is true about Quicksort?

**Answer**

It is an in-place sorting algorithm

**Status :** Correct

**Marks :** 1/1

11. What happens when Merge Sort is applied to a single-element array?

**Answer**

The array remains unchanged and no merging is required

**Status :** Correct

**Marks :** 1/1

12. What is the best sorting algorithm to use for the elements in an array that are more than 1 million in general?

**Answer**

Quick sort.

**Status : Correct**

**Marks : 1/1**

13. What is the main advantage of Quicksort over Merge Sort?

**Answer**

Quicksort requires less auxiliary space

**Status : Correct**

**Marks : 1/1**

14. Which of the following scenarios is Merge Sort preferred over Quick Sort?

**Answer**

When sorting linked lists

**Status : Correct**

**Marks : 1/1**

15. What happens during the merge step in Merge Sort?

**Answer**

Two sorted subarrays are combined into one sorted array

**Status : Correct**

**Marks : 1/1**

16. Which of the following modifications can help Quicksort perform better on small subarrays?

**Answer**

Switching to Insertion Sort for small subarrays

**Status : Correct**

**Marks : 1/1**

17. In a quick sort algorithm, where are smaller elements placed to the

pivot during the partition process, assuming we are sorting in increasing order?

**Answer**

To the left of the pivot

**Status :** Correct

**Marks :** 1/1

18. Which of the following methods is used for sorting in merge sort?

**Answer**

merging

**Status :** Correct

**Marks :** 1/1

19. Consider the Quick Sort algorithm, which sorts elements in ascending order using the first element as a pivot. Then which of the following input sequences will require the maximum number of comparisons when this algorithm is applied to it?

**Answer**

22 25 56 67 89

**Status :** Correct

**Marks :** 1/1

20. The following code snippet is an example of a quick sort. What do the 'low' and 'high' parameters represent in this code?

```
void quickSort(int arr[], int low, int high) {  
    if (low < high) {  
        int pivot = partition(arr, low, high);  
        quickSort(arr, low, pivot - 1);  
        quickSort(arr, pivot + 1, high);  
    }  
}
```

**Answer**

The range of elements to sort within the array

**Status :** Correct

**Marks :** 1/1