QUIZIZZ

NAME : _____

CLASS : _____

DS Unit-1 Test-4
18 Questions

DATE : _____

1. Merge sort uses which of the following techniques to implement sorting?

A Backtracking

B Greedy algorithm

C Divide and conquer

D Dynamic programming

2. What is the best case time complexity of merge sort?

A O(1)

B O(log n)

C O(n)

D O(n log n)

3. What is the worst case time complexity of merge sort?

A O(n log n)

B O(n 2)

C O(n 2 log n)

D O(n log n 2)

4. What is the auxiliary space complexity of merge sort?

A O(1)

B O(log n)

C O(n)

D O(n log n)

5. A stable sorting algorithm

A Does not crash

B Does not run out of memory

C Does not change the sequence of appearance of elements

D Does not exist

?

6. An adaptive sorting algorithm

A Adapts to new inputs

B Takes advantage of already sorted elements

C Takes inputs which are already sorted

D None of the above

7. Which of the following is not an in-place sorting algorithm?

A Merge sort

B Quick sort

C Bubble sort

D Insertion sort

8. Choose the incorrect statement about merge sort from the following:

A It is a comparison based sort

B It is an adaptive algorithm

C It is not an in-place algorithm

D It is a stable algorithm

9. Choose the correct code for a merge sort.

A
```
void merge_sort(int arr[], int left, int right)
{
    if (left > right)
    {

        int mid = (right-left)/2;
        merge_sort(arr, left, mid);
        merge_sort(arr, mid+1, right);

        merge(arr, left, mid, right); //function to merge sorted arrays
    }
}
```

B
```
void merge_sort(int arr[], int left, int right)
{
    if (left < right)
    {

        int mid = left+(right-left)/2;
        merge_sort(arr, left, mid);
        merge_sort(arr, mid+1, right);

        merge(arr, left, mid, right); //function to merge sorted arrays
    }
}
```

C
```
void merge_sort(int arr[], int left, int right)
{
    if (left < right)
    {

        int mid = left+(right-left)/2;
merge(arr, left, mid, right); //function to merge sorted arrays
        merge_sort(arr, left, mid);
        merge_sort(arr, mid+1, right);


    }
}
```

D
```
void merge_sort(int arr[], int left, int right)
{
    if (left < right)
    {

        int mid = (right-left)/2;
merge(arr, left, mid, right); //function to merge sorted arrays
        merge_sort(arr, left, mid);
        merge_sort(arr, mid+1, right);


    }
}
```

10. Quick sort follows the divide-and-conquer strategy.

A True

B False

11. Quick sort is an in-place (internal) sorting algorithm.

A True

B False

12. What is the best case complexity of a quick sort algorithm?

A O(n)

B O(n log n)

C O(n 2)

D O(log n)

13. What is the worst case complexity of a quick sort algorithm?

A O(n)

B O(n log n)

C O($\boldsymbol{n}$ **2**)

D O(log n)

14. Which of the following sorting algorithms is used along with quick sort to sort the sub arrays?

A Merge sort

B Bubble sort

C Insertion sort

D Selection sort

15. How many sub-arrays does the quick sot algorithm divide the entire array into?

A One

B Two

C Three

D Four

16. Which of the below given sorting techniques has the highest best-case runtime complexity?

A Quick sort [nlogn]

B Selection sort [$\boldsymbol{n}$ **2**]

C Insertion sort [n]

D Merge sort [nlogn]

17. The complexity of the sorting algorithm measures the as a function of the number n of items to be sorted.

A Average time

B Running time

C Average-case complexity

D Best-case complexity

18. Partition and exchange sort is

A Merge sort

B Bubble sort

C Quick sort

D Insertions sort

Answer Key

1. c	2. d	3. a	4. c
5. c	6. b	7. a	8. b
9. b	10. a	11. a	12. b
13. c	14. c	15. b	16. b
17. b	18. c		