1) What is Recursion in Java?

a) Recursion is a class

b) Recursion is a process of defining a method that calls other methods repeatedly

**c) Recursion is a process of defining a method that calls itself repeatedly**

d) Recursion is a process of defining a method that calls other methods which in turn call again this method

2) Which of these data types is used by operating system to manage the Recursion in Java?

a) Array

**b) Stack**

c) Queue

d) Tree

3) Which of these will happen if recursive method does not have a base case?

**a) An infinite loop occurs**

b) System stops the program after some time

c) After 1000000 calls it will be automatically stopped

d) None of the mentioned

4) What will be the output of the following Java program?

class recursion

{

int func (int n)

{

int result;

result = func (n - 1);

return result;

}

}

class Output

{

public static void main(String args[])

{

recursion obj = new recursion() ;

System.out.print(obj.func(12));

}

}

a) 0

b) 1

c) Compilation Error

**d) Runtime Error**

5) What will be the output of the following Java program?

class recursion

{

int func (int n)

{

int result;

if (n == 1)

return 1;

result = func (n - 1);

return result;

}

}

class Output

{

public static void main(String args[])

{

recursion obj = new recursion() ;

System.out.print(obj.func(5));

}

}

a) 0

**b) 1**

c) 120

d) None of the mentioned

6) Is there any difference in the speed of execution between linear serach(recursive) vs linear search(lterative)?

a) Both execute at same speed

b) Linear search(recursive) is faster

**c) Linear search(Iterative) is faster**

d) Cant be said

7) Is the space consumed by the linear search(recursive) and linear search(iterative) same?

**a) No, recursive algorithm consumes more space**

b) No, recursive algorithm consumes less space

c) Yes

d) Nothing can be said

8) Linear search(recursive) algorithm used in \_\_\_\_\_\_\_\_\_\_\_\_\_

**a) When the size of the dataset is low**

b) When the size of the dataset is large

c) When the dataset is unordered

d) Never used

9) The array is as follows: 1,2,3,6,8,10. At what time the element 6 is found? (By using linear search(recursive) algorithm)

**a) 4th call**

b) 3rd call

c) 6th call

d) 5th call

10) Which of the following code snippet performs linear search recursively?

a)

for(i=0;i<n;i++)

{

if(a[i]==key)

printf("element found");

}

b) \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

LinearSearch(int[] a, n,key)

{

if(n<1)

return False

if(a[n]==key)

return True

else

LinearSearch(a,n-1,key)

}

c)

LinearSearch(int[] a, n,key)

{

if(n<1)

return True

if(a[n]==key)

return False

else

LinearSearch(a,n-1,key)

}

d)

LinearSearch(int[] a, n,key)

{

if(n<1)

return False

if(a[n]==key)

return True

else

LinearSearch(a,n+1,key)

}

11) What is the recurrence relation for the linear search recursive algorithm?

a) T(n-2)+c

b) 2T(n-1)+c

**c) T(n-1)+c**

d) T(n+1)+c

12) Given an input arr = {2,5,7,99,899}; key = 899; What is the level of recursion?

a) 5

b) 2

**c) 3**

d) 4

13) What is the time complexity of binary search with iteration?

a) O(nlogn)

**b) O(logn)**

c) O(n)

d) O(n2)

14) Which of the following is not a stable sorting algorithm?

a) Insertion sort

**b) Selection sort**

c) Bubble sort

d) Merge sort

15) Which of the following is not an in-place sorting algorithm?

a) Selection sort

b) Heap sort

c) Quick sort

**d) Merge sort**

16) Running merge sort on an array of size n which is already sorted is

a) O(n)

**b) O(nlogn)**

c) O(n2)

d) None

17) Which of the following is not a noncomparison sort?

a) Counting sort

b) Bucket sort

c) Radix sort

**d) Shell sort**

18) If the given input array is sorted or nearly sorted, which of the following algorithm gives the best performance?

**a) Insertion sort**

b) Selection sort

c) Quick sort

d) Merge sort

19) Given a number of elements in the range [0….n3]. which of the following sorting algorithms can sort them in O(n) time?

a) Counting sort

b) Bucket sort

**c) Radix sort**

d) Quick sort

20) Counting sort performs …………. Numbers of comparisons between input elements.

**a) 0**

b) n

c) nlogn

d) n2

21) Which of the following algorithm design technique is used in the quick sort algorithm?

a) Dynamic programming

b) Backtracking

**c) Divide-and-conquer**

d) Greedy method

22) For merging two sorted lists of size m and n into sorted list of size m+n, we require comparisons of

a) O(m)

b) O(n)

**c) O(m+n)**

d) O(logm + logn)

23) Using asymptotic analysis, we can very well conclude the \_\_\_\_\_\_\_\_\_\_ scenario of an algorithm.

a) best case

b) average case

c) worst case

**d) best case, average case, and worst case**

24) \_\_\_\_\_\_\_\_\_\_ is the formal way to express the upper bound of an algorithm's running time.

a) Omega Notation

b) Theta Notation

**c) Big Oh Notation**

d) All of the above

25) Ο(log n) is?

a) constant asymptotic notations

**b) logarithmic asymptotic notations**

c) polynomial asymptotic notations

d) quadratic asymptotic notations

26) Asymptotic analysis is \_\_\_\_\_\_\_ bound.

a) output

**b) input**

c) outer

d) inner

27) What is the disadvantage of selection sort?

a) It requires auxiliary memory

**b) It is not scalable**

c) It can be used for small keys

d) None of the mentioned

28) Which of the following is not O(n2)?

a) (15^10) \* n + 12099

b) n^1.98

**c) n^3/(sqrt(n))**

d) (2^20) \* n

29) When determining the efficiency of algorithm, the space factor is measured by \_\_\_\_

**a) Counting the maximum memory needed by the algorithm**

b) Counting the minimum memory needed by the algorithm

c) Counting the average memory needed by the algorithm

d) Counting the maximum disk space needed by the algorithm

30)When determining the efficiency of algorithm the time factor is measured by\_\_\_\_\_\_

a) Counting microseconds

**b) Counting the number of key operations**

c) Counting the number of statements

d) Counting the kilobytes of algorithm