SRM INSTITUTE OF SCIENCE AND TECHNOLOGY, RAMAPURAM CAMPUS

COMPUTER SCIENCE AND ENGINEERING

QUESTION BANK

18CSC204J DESIGN AND ANALYSIS OF ALGORITMS

UNIT 1

Introduction-Algorithm Design, Fundamentals of Algorithms, Correctness of algorithm, Time complexity analysis, Insertion sort-Line count, Operation count, Algorithm Design paradigms, Designing an algorithm, And its analysis-Best, Worst and Average case, Asymptotic notations Based on growth functions. O,O,Θ, ω, Ω Mathematical analysis, Induction, Recurrence relations , Solution of recurrence relations, Substitution method, Solution of recurrence relations, Recursion tree, Solution of recurrence relations, Examples

PART A

1. _____is the first step in solving the problem

A. Understanding the Problem
B. Identify the Problem
C. Evaluate the Solution
D. Coding the Problem

Answer: - B

2. While solving the problem with computer the most difficult step is _____.

A. describing the problem
B. finding out the cost of the software
C. writing the computer instructions
D. testing the solution

Answer:- C

3. _____ solution requires reasoning built on knowledge and experience

A. Algorithmic Solution
B. Heuristic Solution
C. Random Solution
D. Brute force Solution

Answer: - B

4. The correctness and appropriateness of _____solution can be checked very easily.

A. algorithmic solution
B. heuristic solution
C. random solution
D. Brute force Solution

Answer:- A

5. 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

Answer: - A

6. The elements of an array are stored successively in memory cells because

A. by this way computer can keep track only the address of the first element and the addresses of other elements can be calculated
B. the architecture of computer memory does not allow arrays to store other than serially
C. Either A or B
D. Both A and B

Answer: - A

7. The hierarchy of operations is denoted as _____.

I. +, - II. Power III. *, / IV. \, MOD

A. I, II, III, IV
B. II, IV, III, I
C. IV, I, III, II
D. II, III, IV, I

Answer:- B

8. What is the time complexity of following code:

```
int a = 0, i = N;
while (i > 0)
{
a += i;
i /= 2;
}
```

A. O(N)
B. O(Sqrt(N))
C. O(N / 2)
D. O(log N)

Answer: - D

9. Two main measures for the efficiency of an algorithm are

A. Processor and memory

B. Complexity and capacity

C. Time and space

D. Data and space

Answer: - C

10. What does the algorithmic analysis count?

A. The number of arithmetic and the operations that are required to run the program

B. The number of lines required by the program

C. The number of seconds required by the program to execute

D. None of these

Answer:- A

11. An algorithm that indicates the amount of temporary storage required for running the algorithm, i.e., the amount of memory needed by the algorithm to run to completion is termed as_____.

A. Big Theta θ (f)

B. Space complexity

C. Big Oh O (f)

D. Time Complexity

Answer B

12. Consider a linked list of n elements. What is the time taken to insert an element after an element pointed by some pointer?

A. (1)

B. (n)

C. (log2 n)

D. (n log2 n)

Answer A

13. If the address of A[1][1] and A[2][1] are 1000 and 1010 respectively and each element occupies 2 bytes then the array has been stored in order.

A. row major

B. column major

C. matrix major

D. none of these

Answer A

14. The time factor when determining the efficiency of algorithm 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

Answer B

15. Time complexities of three algorithms are given. Which should execute the slowest for large values of N?

A. (n log n)

B. O(n)

C. O(log n)

D. O(n2)

Answer B

16. Which one of the following is the tightest upper bound that represents the number of swaps required to sort n number using selection sort?

A. (log n)

B. O(n)

C. (n log n)

D. $O(n^2)$

Answer B

17. How many comparisons are needed for linear Search array when elements are in order in best case?

A. 1

B. n

C. n+1

D. n-1

Answer A

18. The complexity of Bubble sort algorithm is_____

A. O(n)

B. O(log n)

C. O(n2)

D. O(n log n)

Answer : C

19. What is the time complexity of following code:

```
int a = 0, i = N;
while (i > 0)
{
a += i;
i /= 2;
}
```

A. O(N)

B. O(Sqrt(N))

C. O(N / 2)

D. O(log N)

Answer D

20. **Which of the given options provides the increasing order of asymptotic complexity of functions f1, f2, f3 and f4?**

f1(n) = 2^n **f3(n) = nLogn**

f2(n) = n^(3/2) **f4(n) = n^(Logn)**

A. f3, f2, f1, f4

B. f2, f3, f1, f4

C. f2, f3, f4, f1

D. f3, f2, f4, f1

Answer is: D

21. How much number of comparisons is required in insertion sort to sort a file if the file is sorted in reverse order?

A. N^2 C. N-1

B. N D. N/2

Answer A

22. The worst-case occur in linear search algorithm when .....

A. Item is somewhere in the middle of the array
B. Item is not in the array at all
C. Item is the last element in the array
D. Item is the last element in the array or item is not there at all

Answer D

23. What is the time complexity of fun()?

```
int fun(int n)
{
  int count = 0;
  for (int i = 0; i < n; i++)
     for (int j = i; j > 0; j--)
        count = count + 1;
  return count;
}
```

A. Theta (n) C. Theta (n*Logn)

B. Theta (n^2) D. Theta (nLognLogn)

Answer : B

24. The time complexity of the following C function is (assume n > 0)

```
(int recursive (mt n)
{
   if (n == 1)
     return (1);
   else
     return (recursive (n-1) + recursive (n-1));
```

}

A. O(n)
B. O(nlogn)
C. $O(n^2)$
D. $O(2^n)$

Answer D

25. **A function in which f(n) is Ω(g(n)), if there exist positive values k and c such that f(n)>=c*g(n), for all n>=k. This notation defines a lower bound for a function f(n):**

A. Big Omega Ω (f)
B. Big Theta θ (f)
C. Big Oh O (f)
D. Big Alpha α (f)

Answer A

26. The concept of order Big O is important because_____

A. It can be used to decide the best algorithm that solves a given problem
B. It determines the maximum size of a problem that can be solved in a given amount of time
C. It is the lower bound of the growth rate of algorithm
D. Both A and B

Answer A

27. The upper bound on the time complexity of the nondeterministic sorting algorithm is

A. O(n)
B. O(n log n)
C. O(1)
D. O(log n)

Answer: A

28. In the analysis of algorithms, what plays an important role?

A. Text Analysis
B. Growth factor
C. Time
D. Space

Answer: B

29. Which one of the following correctly determines the solution of the recurrence relation given below with T(1) = 1 and $T(n)= 2T(n/4) + n^{1/2}$

A. $O(n^2)$
B. O(n)
C. $O(n^{1/2} \log n)$
D. O(log n)

Answer C

30. What is the time complexity of recursive function given below:

$T(n)= 4T(n/2) + n^2$

A. $O(n^2)$

B. $O(n)$

C. $O(n^2 \log n)$

D. $O(n \log n)$

Answer C

PART B

1 . What is an Algorithm?
2 . Give the notion of an algorithm.
3 . Design an algorithm for computing gcd(m,n) using Euclid's algorithm.
4 . Design an algorithm to compute the area and circumference of a circle.
5 . Differentiate Sequential and Parallel Algorithms.
6 . Write the process for design and analysis of algorithm.
7 . What are the fundamentals steps for design and analysis of an algorithm?
8 . Compare Exact and Approximation algorithm.
9 . What is an Algorithm Design Technique?
10 . Define Pseudo code.
11 . Define Flowchart.
12 . Prove the correctness of an algorithm's.
13 . Define algorithm validation.
14 . What is validation and program verification?
15 . Define program proving and program verification.
16 . Write the characteristics of an algorithm.
17 . What is the Efficiency of algorithm?
18 . What is time and space complexity?
19 . What is generality of an algorithm?
20 . What is algorithm's Optimality?
21 . Write an algorithm to find the number of binary digits in the binary representation of a positive decimal integer. Answer : log n (base 2) + 1
22 . What are the types of problems in algorithm?
23 . How will you measure input size of algorithms?
24 . What is the average case complexity of linear search algorithm?
25 . Differentiate searching and sorting algorithm.
26 . What are combinatorial problems?
27 . Define a graph and its type.
28 . Define performance analysis.
29 . What do you mean by Worst case-Efficiency of an algorithm?
30 . What do you mean by Best case-Efficiency of an algorithm?
31 . Define the Average-case efficiency of an algorithm.
32 . What do you mean by Amortized efficiency? https://www.geeksforgeeks.org/analysis-algorithm-set-5-amortized-analysis-introduction/
33 . How to analyze an algorithm framework?

34 . How to measure the algorithm's efficiency?
35 . What is called the basic operation of an algorithm?
36 . How to measure an algorithm's running time?
37 . Define time and space complexity.
38 . Write an algorithm for adding 'n' natural numbers and find the time and space required by that algorithm.
39 . Define order of growth.
40 . What is meant by linear search?
41 . Compare the two functions 2^n and n^2 for various values of n. Determine when the second function will become the same, smaller and larger than the first function.
42 . What are the properties of big-Oh notation?
43 . Define Big oh notation.
44 . Define little Oh and Omega notations.
45 . Define Ω notation.
46 . Define Θ – notation.
47 . What is the use of Asymptotic Notations?
48 . What are the properties of asymptotic notations?
49 . Mention the general plan for analyzing time efficiency of Non recursive algorithms.
50 . Define recursive and non – recursive algorithm.
51 . What is recurrence equation?
52 . Define Recurrence relation with an example.
53 . Give the time complexity 1+3+5+7+….+999.
54 . Compare order of growth n(n-1)/2 and n^2 .
55 . Find the order of growth of the following sums.

$$\sum_{i=1}^{n-1} (i^2 + 1)^2$$

56 . Solve the following recurrence relations.
X(n)=x(n-1) + 5 for n>1, x(1) = 0
57 . Consider the following algorithm

```
S=0
for =1 to n do
S=S+i
return i
```

What does this algorithm compute? How many times is the basic operation executed?
58 . Design an algorithm to compute the area and Circumference of a circle.
59 . The (log n)th smallest number of n unsorted numbers can be determined in O(n) average-case time.
60 . Write the recursive Fibonacci algorithm and its recurrence relation.

PART C

1 . Describe the steps in analyzing & coding an algorithm.

2 . Enumerate the problem types used in the design of algorithm.

3 . What are the steps that need to be followed while designing and analyzing algorithm?

4 . Explain the fundamental of algorithmic problem solving. https://snscourseware.org/snscenew/files/1579691819.pdf

5 . Use the most appropriate notation to indicate the time efficiency class of sequential algorithm in the worst case, best case and the average case.

6 . Consider the following algorithm for the searching problem.

Algorithm:

```
Linear search (A[0,...n-1],key)
//Searches an array for a key value by linear search
//Input: Array A[0..n-1] of values and a key value to search
//Output: Returns index if search is successful
for i<-0 to n-1 do
        if(key==A[i])
        return i
```

7 . Explain some of the problem types used in the design of algorithm.

8 . Define time complexity and space complexity. Write an algorithm for adding ‘n’ natural numbers and find the time and space required by that algorithm.

9 . Explain the general framework for analyzing the efficiency of algorithm.

10 . Write the Insertion Sort algorithm and estimate its running time.

11 . What is space complexity? With an example, explain the components of fixed and Variable part in space complexity?

12 . Show how to implement a stack using two queues. Analyze the running time of stack operations.

13 . Discuss the properties of asymptotic notations. https://www.geeksforgeeks.org/properties-of-asymptotic-notations/#:~:text=Assuming%20f(n)%2C%20g,f(n)%20%E2%89%A4%20c2.

14 . Explain the various asymptotic notations used in algorithm design. With an Example

15 . Give the definition and graphical representation of O notations.

16 . Define asymptotic notations. Distinguish between Asymptotic notation and conditional asymptotic notation.

17 . Prove that for any two functions f(n) and g(n), we have

f(n) = θ (g(n)) if and only if f(n) = O (g(n)) and f(n) = Ω(g(n)).

18 . Write the linear search algorithm and analyze for its best worst and average case time Complexity.

19 . Discuss about recursive and non-recursive algorithms with example.

20 . What is the general plan for time efficiency of recursive algorithm and find the number of binary digits in the binary representation of positive decimal integer find recurrence relation and complexity.

21 . State the general plan for analyzing the time efficiency of non-recursive algorithms and explain with an example.

22 . Compare the order of the growth of the following.

i)(1/2) n(n-1) and n^2

ii)$\log_2 n$ and $\sqrt{n}$

iii)n! and 2^n

23 . Find the closest asymptotic tight bound by solving the recurrence equation T(n)=8T(n/2)+n2 with (T(1)=1) using Recursion tree method. [Assume T(1) € θ (1)]

24 . Give an algorithm to check whether all the elements in a given array of n-elements are distinct, find the worst case complexity of the same.

25 . Explain the towers of Hanoi problem and solve it using recursion

26 . Prove the time complexity of the matrix multiplication is $O(n^3)$

27 . Define recurrence equation and explain how solving recurrence equations are done.

28 . Solve the following recurrence relations.

I. x(n)=x(n-1)+5 for n>1, x(1)=0

II. x(n)=3x(n-1)for n>1, x(1)=4

III. x(n)=x(n-1)+n for n>0, x(0)=0

IV. x(n)=x(n/2)+n for n>1, x(1)=1 (solve for $n=2^k$)

V. x(n)=x(n/3)+1 for n>1, x(1)=1 (solve for $n=3^k$)

29 . Solve the following recurrence relations.

I. x(n)=x(n-1)+n for n>0,x(0)=0

II. x(n)=x(n/2)+n for n>1, x(1)=1 (solve for $n=2^k$)

III. x(n)=3x(n-1) for n>1,x(1)=4

30 . Suppose W satisfies the following recurrence equation and base case (where c is constant)

: W(n)=c.n+W(n/2) and W(1)=1. What is the asymptotic order of W(n). With a suitable example, explain the method of solving recurrence equations.

31 . Consider the following recursion algorithm Min1(A[0 -------n-1])

If n=1 return A[0]

Else temp = Min1(A[0…….n-2]) If temp <= A[n-1] return temp

Else

Return A[n-1]

What does this algorithm compute?

32 . Consider the following algorithm.

```
Algorithm :
Sum(n)
// A non negative integer n
S <- 0
for i <- 1 to n do
S<-S+i
Return S
```

i. What does this algorithm compute?
ii. What is its basic operation?
iii. How many times is the basic operation executed?
iv. What is the efficiency class of this algorithm?
v. Suggest an improved algorithm and indicate its efficiency class. If you cannot do it, try to prove that it cannot be done.

33 . Setup a recurrence relation for the algorithms basic operation count and solve it.

34 . Derive the recurrence relation for Fibonacci series algorithm; also carry out the time complexity analysis.

35 .Give the non recursive algorithm for finding the value of the largest element in a list of n numbers.