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.
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?
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.

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.

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.

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. O(Sqrt(N)) C. IV, I, III, II D. II, III, IV, I

B. II, IV, III, 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) 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) Answer B

B. Space complexity C. Big Oh O (f) D. Time Complexity

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)

Answer A

C. (log2 n) D. (n log2 n)

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)

Answer B

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

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

A. 1

B. n

Answer A

C. n+1 D. n-1

18. The complexity of Bubble sort algorithm is_____

Answer : C

A. O(n)

B. O(log n)

C. O(n2)

D. O(n log n)

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
f2(n) = n^(3/2) A. f3, f2, f1, f4 B. f2, f3, f1, f4 C. f2, f3, f4, f1 D. f3, f2, f4, f1 Answer is: D
f3(n) = nLogn
f4(n) = n^(Logn)

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
B. N
Answer A
C. N-1 D. N/2

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) B. Theta (n^2) Answer : B
C. Theta (n*Logn) D. Theta (nLognLogn)

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));
```

}

C. $O(n^2)$ D. $O(2^n)$

A. O(n)

B. O(nlogn) 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)

Answer: A

C. O(1)

D. O(log n)

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

A. Text Analysis B. Growth factor Answer: B

C. Time D. Space

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. $O(n)$

Answer C

PART B

1 . What is an Algorithm?

2 . Give the notion of an algorithm.

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

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.

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?

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.

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.

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.

UNIT II: DIVIDE AND CONQUER

Introduction-Divide and Conquer, Maximum Sub array Problem
Binary Search, Complexity of binary search
Merge sort, Time complexity analysis
Quick sort and its Time complexity analysis, Best case, Worst case, Average case analysis
Strassen's Matrix multiplication and its recurrence relation, Time complexity analysis of Merge sort
Largest sub-array sum, Time complexity analysis of Largest sub-array sum
Master Theorem Proof, Master theorem examples
Finding Maximum and Minimum in an array, Time complexity analysis-Examples
Algorithm for finding closest pair problem, Convex Hull problem

PART-A

1.) Partition and exchange sort is_____
A. quick sort
B. tree sort
C. heap sort
D. bubble sort
ANSWER: A

2) Which of the following is not the required condition for binary search algorithm?
A. The list must be sorted
B. There should be the direct access to the middle element in any sub list
C. There must be mechanism to delete and/or insert elements in list.
D. Number values should only be present
ANSWER: C

3) Which of the following sorting algorithm is of divide and conquer type?
A. Bubble sort
B. Insertion sort
C. Merge sort
D. Selection sort
ANSWER: C

4) _____order is the best possible for array sorting algorithm which sorts n item.
A. O (n logn)
B. O (n2)
C. O (n+logn)
D. O (logn)
ANSWER: C

5) The complexity of merge sort algorithm is _____
A. O (n)
B. O (logn)

C. O (n2)
D. O (n logn)
ANSWER: D

6) Binary search algorithm cannot be applied to _____
A. sorted linked list
B. sorted binary trees
C. sorted linear array
D. pointer array
ANSWER: A

7) Which of the following is not a limitation of binary search algorithm?
A. must use a sorted array
B. requirement of sorted array is expensive when a lot of insertion and deletions are needed
C. there must be a mechanism to access middle element directly
D. binary search algorithm is not efficient when the data elements more than 1500.
ANSWER: D

8) Which of the following is an external sorting?
A. Insertion Sort
B. Bubble Sort
C. Merge Sort
D. Tree Sort
ANSWER: B

9) Merging k sorted tables into a single sorted table is called _____
A. k way merging
B. k th merge
C. k+1 merge
D. k-1 merge
ANSWER: A

10) The operation that combines the element is of A and B in a single sorted list C with n=r+s element is called _____
A. Inserting
B. Mixing
C. Merging
D. Sharing
ANSWER: C

11) Which of the following is a stable sorting algorithm?
a) Merge sort
b) typical in-place quick sort
c) Heap sort
d) Selection sort
ANSWER: A

12) Which of the following is not an in-place sorting algorithm?
a) Selection sort
b) Heap sort
c) Quick sort
d) Merge sort
ANSWER: D

13)The time complexity of a quick sort algorithm which makes use of median, found by an O(n) algorithm, as pivot element is
a) $O(n^2)$
b) O(nlogn)
c) O(nloglogn)
d) O(n)
ANSWER: B

14) 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
ANSWER: C

15) Merge sort uses
a) Divide-and-conquer
b) Backtracking
c) Heuristic approach
d) Greedy approach
ANSWER: A

16)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)
ANSWER: C

17) The running time of Strassen's algorithm for matrix multiplication is
(A) $\Theta(n)$ (B) $\Theta(n^3)$ **(**C) $\Theta(n^2)$ (D) $\Theta(n^{2.81})$
ANSWER: D

18) The Stassen's algorithm's time complexity is
(A) 0(n) (B) 0(n^2) (C) 0(n^2.80) (D) 0(n^2.81)
ANSWER: C

19) Which algorithm is used for matrix multiplication?
a. Simple algorithm
b. Specific algorithm
c. Strassen algorithm
d. Addition algorithm
ANSWER: C

20) Which algorithm is a divided and conquer algorithm that is asymptotically faster:
a. Simple algorithm
b. Specific algorithm
c. Strassen algorithm
d. Addition algorithm
ANSWER: C

21) Which algorithm is named after Volker Strassen
a. Strassen algorithm
b. Matrix algorithm
c. Both
d. None of these
ANSWER: A

22) Which of the following algorithms is NOT a divide & conquer algorithm by nature?
(A) Euclidean algorithm to compute the greatest common divisor
(B) Heap Sort
(C) Closest pair problem
(D) Quick Sort
Answer: B

23). what is the average case time complexity of merge sort?
a) O (n log n)
b) O (n2)
c) O (n2 log n)
d) O (n log n2)
ANSWER: A

24). which of the following method is used for sorting in merge sort?
a) Merging
b) Partitioning
c) Selection
d) Exchanging
ANSWER: A

25) Which of the following is not a stable sorting algorithm?
a) Quick sort

b) Cocktail sort
c) Bubble sort
d) Merge sort
ANSWER: A

26) What is the runtime efficiency of using brute force technique for the closest pair problem?
a) O (N)
b) O (N log N)
c) O (N2)
d) O (N3 log N)
ANSWER: C

27) What is the basic operation of closest pair algorithm using brute force technique?
a) Euclidean distance
b) Radius
c) Area
d) Manhattan distance
ANSWER: A

28) _____ is a method of constructing a smallest polygon out of n given points.
a) Closest pair problem
b) Quick hull problem
c) Path compression
d) union-by-rank
ANSWER: B

29) Find the maximum sub-array sum for the given elements.
{-2, -1, -3, -4, -1, -2, -1, -5, -4}
a) -3
b) 5
c) 3
d) -1
ANSWER: D

30) Master's theorem is used for?
a) Solving recurrences
b) Solving iterative relations
c) Analyzing loops
d) Calculating the time complexity of any code
ANSWER: A

PART-B

1) Determine the efficiency of divide and conquer algorithms.
2) Explain and analyze the merge sort algorithm.
3) How quick sort can be improved?
4) Explain the binary searching algorithm in detail, with an example. Show the worst case efficiency of binary search is in Θ (log n)
5) Explain the divide and conquer strategy with examples.
6) Apply quick sort to sort the list E, X, A, M, P, L, E in alphabetical order. Draw the tree of the recursive calls made.
    i. Write the best case input for the quick sort
    ii. Find the best case time efficiency for the quick sort
    iii. Are the arrays made up of all equal elements the worst case input, the best case input or neither?
7) Write a pseudo code for divide & conquer algorithm for finding the position of the largest element in an array of numbers.
8) Write a quick sort algorithm and derive the worst case and average case complexity class of this algorithm.
9) Discuss the efficiency of quick sort algorithm.
10) Explain how the merge sort can be viewed as a recursive application of the Divide and conquer methodology. Suggest a pseudo code for merge sort and analyze its complexities. Trace its application to the following data set 9, 4, 3,8,6,2,1,5,7.
11) Explain Strassen's matrix multiplication. Evaluate its efficiency.
12) Discuss about the Quick Hull Algorithm.
13) Explain in Detail about the Closest pair Algorithm
14) Explain in detail about min-max problem using divide and conquer and derive its time complexity.
15) Discuss the advantages and disadvantages of Divide and Conquer Algorithm?

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

RAMAPURAM CAMPUS

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

QUESTION BANK

18CSC204J DESIGN AND ANALYSIS OF ALGORITM (REGULATION 2018)

UNIT III

Introduction-Greedy and Dynamic Programming, Examples of problems that can be solved by using greedy and dynamic approach Huffman coding using greedy approach, Comparison of brute force and Huffman method of encoding Knapsack problem using greedy approach, Complexity derivation of knapsack using greedy Tree traversals, Minimum spanning tree - greedy, Kruskal's algorithm - greedy Minimum spanning tree - Prim's algorithm, Introduction to dynamic programming 0/1 knapsack problem, Complexity calculation of knapsack problem Matrix chain multiplication using dynamic programming, Complexity of matrix chain multiplication Longest common subsequence using dynamic programming, Explanation of LCS with an example Optimal binary search tree (OBST)using dynamic programming, Explanation of OBST with an example

PART A

1. ---------- is a Boolean-valued function that determines whether x can be included into the solution vector

    a) Overlapping subproblems
    b) Fleasible solution
    c) Memoization
    d) Greedy
2. Trees with edge with weights are called ----------
    a) weighted tree
    b) unweighted tree
    c) bruteforce
    d) Greedy
3. ---------- is to determine an optimal placement of booster
    a) Weighted tree
    b) Vertex
    c) Tree Vertex Splitting Problem (TVSP)
    d) Greedy
4. The order in which TVS visits that computes the delay values of the nodes of the tree is called the----------.
    a)treeorder
    b) inorder
    c) preorder

d) postorder

5. Algorithm TVS takes ---------- time, where n is the number f nodes in the tree
   a)O(N)
   b)Ω(n log n)
   c)O (n2log n)
   d)O(n log n)
6. ---------- is a greedy method to obtain a minimum-cost spanning tree builds this tree edge by edge
   a)Prism's algorithm
   b)Dynamic algorithm
   c)Greedy algorithm
   d)Dynamic algorithm

7. Kruskal's algorithm (choose best non-cycle edge) is better than Prim's (choose best tree edge) when the graph has relatively few edges

   a)True
   b)False
8. Two sorted files containing n and m records respectively could be merged together to obtain one sorted file in time ----------.

   a) Ω(n log n)
   b) **O(n+m)**
   c) O (n2log n)
   d)O(n log n)

9. The two-way merge pattern scan be represented by----------
   a) Weighted tree
   b) Vertex
   c) binary merge tree
   d) Greedy
10. What algorithm technique is used in the implementation of Kruskal solution for the MST?
   a) greedy technique
   b) divide-and-conquer technique
   c) dynamic programming technique
   d) the algorithm combines more than one of the above techniques

11. The function Tree of Algorithm uses the ------- stated to obtain a two-way merge tree for n file
   a) divide-and-conquer technique
   b) greedy rule
   c) dynamic programming technique
   d) the algorithm combines more than one of the above techniques

12. A decode tree is a---------- in which external nodes represent messages.
   a) minimum spanning tree
   b) B tree
   c) binary tree

d) AVL tree

13. The ----------in the code word for a message determine the branching needed at each level of the decode tree to reach the correct external node.
    a) binary bits
    b) decoder
    c) encoder
    d) binary bytes

14. The cost of decoding a --------is proportional to the number of bits in the code
    a) binary bits
    b) code word
    c) data
    d) binary bytes

15. What is the edges on the shortest paths from a vertex v to all remaining vertices in a connected undirected graph G form a spanning tree of G is called?
    a) MST
    b) shortest-path spanning tree
    c) binary tree
    d) AVL tree

16. ----------is an algorithm design method that can be used when the solution to a problem can be viewed as the result of a sequence of decisions.
    a) Dynamic Programming
    b) Greedy method
    c) Huffman coding
    d) Tree traversal

17. What is an another important feature of the dynamic programming approach that optimal solutions are retained so as to avoid recomputing their values.
    a) Dynamic Programming
    b) Greedy method
    c) Huffman coding
    d) Tree traversal

18. --------- often drastically reduces the amount of enumeration by avoiding the enumeration of some decision sequences that cannot possibly be optimal.
    a) Dynamic Programming
    b) Greedy method
    c) Huffman coding
    d) Tree traversal

19. In the ----------only one decision sequence is ever generated.
    a) Dynamic Programming
    b) Greedy method
    c) Huffman coding
    d) Tree traversal

20. Dynamic programming algorithms solve the---------- to obtain a solution to the given problem instance

a) optimistic
b) Greedy method
c) Huffman coding
d) recurrence

21. A dynamic programming formulation for a k-stage graph problem is obtained by first noticing that every s to t path is the result of a sequence of ---------- decision.
    a) k
    b) k-1
    c) k-2
    d) 2k

22. Which of the following problems is NOT solved using dynamic programming?
    a) 0/1 knapsack problem
    b) Matrix chain multiplication problem
    c) Edit distance problem
    d) Fractional knapsack problem

23. The problem of ---------is to identify a minimum-cost sequence of edit operations that will transform X into Y.
    a) 0/1 knapsack problem
    b) Matrix chain multiplication problem
    c) Edit distance problem
    d) string editing

24. In Knapsack problem, the best strategy to get the optimal solution, where Pi, Wi is the Profit, Weight associated with each of the Xi^{th} object respectively is to
    a)Arrange the values Pi/Wi in ascending order
    b)Arrange the values Pi/Xi in ascending order
    c)Arrange the values Pi/Wiin descending order
    d)Arrange the values Pi/Xiin descending order

25. Greedy job scheduling with deadlines algorithms' complexity is defined as
    a)O(N)
    b)Ω(n log n)
    c)O (n2log n)
    d)O(n log n)

26. In Huffman coding, data in a tree always occur?
    a) roots
    b) leaves
    c) left sub trees
    d) right sub trees

27. The multistage graph problem can also be solved using the ----------
    a) backward approach

b) forward approach
c) brute force approach
d) right sub trees

28. The all-pairs ----------problem is to determine a matrix A such that A(i,j)is the length of a shortest path from i to j.
    a) backward approach
    b) forward approach
    c) brute force approach
    d) shortest-path

29. Which of the following methods can be used to solve the Knapsack problem?

    a) Brute force algorithm
    b) Recursion
    c) Dynamic programming
    d) Brute force, Recursion and Dynamic Programming

30. The inorder and preorder traversal of a binary tree are d b e a f c g and a b d e c f g, respectively. The postorder traversal of the binary tree is:
    a) d e b f g c a
    b) e d b g f c a
    c) e d b f g c a
    d) d e f g b c a

31. Which of the following pairs of traversals is not sufficient to build a binary tree from the given traversals?
    a) Preorder and Inorder
    b) Preorder and Postorder
    c) Inorder and Postorder
    d) Inorder and levelorder

32. Consider the following C program segment

```
struct CellNode
{
 struct CelINode *leftchild;
 int element;
 struct CelINode *rightChild;
}

int Dosomething(struct CelINode *ptr)
{
   int value = 0;
   if (ptr != NULL)
   {
     if (ptr->leftChild != NULL)
       value = 1 + DoSomething(ptr->leftChild);
     if (ptr->rightChild != NULL)
       value = max(value, 1 + DoSomething(ptr->rightChild));
```

```
    }
    return (value);
}
```

The value returned by the function DoSomething when a pointer to the root of a non-empty tree is passed as argument is

a) The number of leaf nodes in the tree
b) The number of nodes in the tree
c) The number of internal nodes in the tree
d) The height of the tree

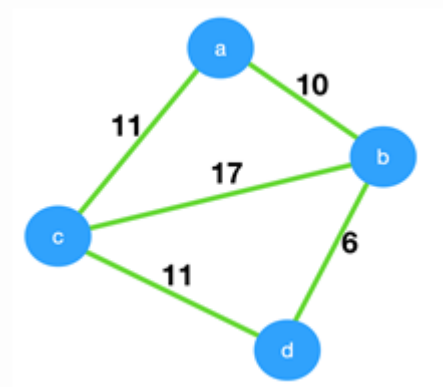
33. Given items as {value, weight} pairs {{60, 20}, {50, 25}, {20, 5}}. The capacity of knapsack=40. Find the maximum value output assuming items to be divisible and nondivisible respectively.

a) 100,80
b) 110,70
c) 130,110
d) 110,80

34. Given items as {value, weight} pairs {{40, 20}, {30, 10}, {20, 5}}. The capacity of knapsack=20. Find the maximum value output assuming items to be divisible

a) 60
b) 80
c) 100
d) 40

35. Consider the given graph.



What is the weight of the minimum spanning tree using the Prim's algorithm, starting from vertex a?

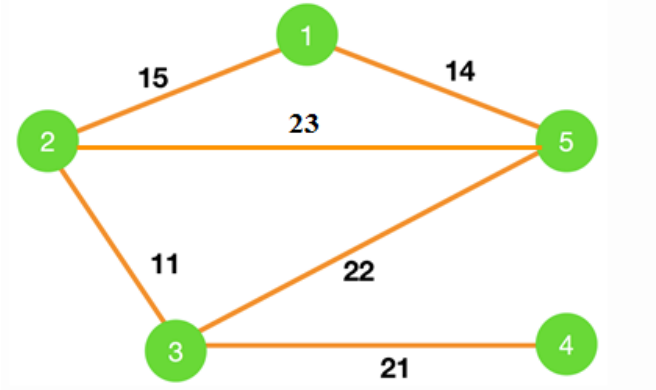
a) 23
b) 28
c) 27
d) 11

36. Worst case is the worst case time complexity of Prim's algorithm if adjacency matrix is used?

a) O(log V)

b) O(V^2)
c) O(E^2)
d) O(V log E)

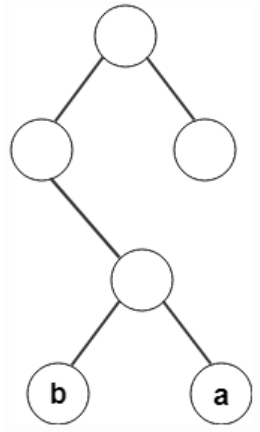
37. Consider the graph shown below.



Which of the following edges form the MST of the given graph using Prim'a algorithm, starting from vertex 4.

a) (4-3)(5-3)(2-3)(1-2)
b) (4-3)(3-5)(5-1)(1-2)
c) (4-3)(3-5)(5-2)(1-5)
d) (4-3)(3-2)(2-1)(1-5)

38. From the following given tree, what is the code word for the character 'a'?



a) 011
b) 010
c) 100
d) 101

39. What will be the cost of the code if character c_i is at depth d_i and occurs at frequency f_i?

a) $c_i f_i$
b) $\int c_i f_i$
c) $\sum f_i d_i$

d) $f_i d_i$

40. What is the running time of the Huffman encoding algorithm?

a) O(C)
b) O(log C)
c) O(C log C)
d) O(N log C)

41. The weighted array used in TVS problems for the following binary tree is _____

a) [1,2,3,0,0,4,0,5,6]
b) [1,2,3,0,0,4,0,5,0,0,0,0,6]
c) [1,2,3,4,5,6]
d) [1,2,3,0,0,4,5,6]

42. What is the time complexity of the brute force algorithm used to find the longest common subsequence?

a) O(n)
b) $O(n^2)$
c) $O(n^3)$
d) $O(2^n)$

43. Find the longest increasing subsequence for the given sequence:
{10, -10, 12, 9, 10, 15, 13, 14}

a) {10, 12, 15}
b) {10, 12, 13, 14}
c) {-10, 12, 13, 14}
d) {-10, 9, 10, 13, 14}

44. What is the space complexity of the following dynamic programming implementation used to find the length of the longest increasing subsequence?

```
#include<stdio.h>
int longest_inc_sub(int *arr, int len)
{
    int i, j, tmp_max;
    int LIS[len];  // array to store the lengths of the longest increasing subsequence
    LIS[0]=1;
    for(i = 1; i < len; i++)
    {
        tmp_max = 0;
            for(j = 0; j < i; j++)
            {
                if(arr[j] < arr[i])
                {
                        if(LIS[j] > tmp_max)
                         tmp_max = LIS[j];
```

```
                }
            }
                LIS[i] = tmp_max + 1;
        }
        int max = LIS[0];
        for(i = 0; i < len; i++)
                if(LIS[i] > max)
                  max = LIS[i];
        return max;
}
int main()
{
        int arr[] = {10,22,9,33,21,50,41,60,80}, len = 9;
        int ans = longest_inc_sub(arr, len);
        printf("%d",ans);
        return 0;
}
```

a) O(1)
b) O(n)
c) O(n2)
d) O(nlogn)

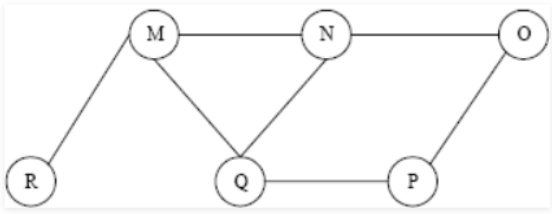
45. The Breadth First Search algorithm has been implemented using the queue data structure. One possible order of visiting the nodes of the following graph is

a) O(1)
b) O(n)
c) O(n2)
d) O(nlogn)

46. Uniform-cost search expands the node n with the _____

a) Lowest path cost
b) Heuristic cost
c) Highest path cost
d) Average path cost

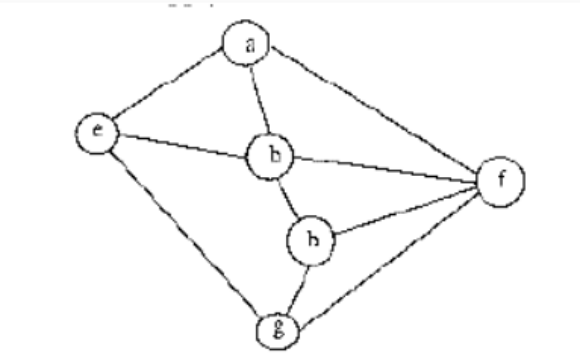
47. The Breadth First Search algorithm has been implemented using the queue data structure. One possible order of visiting the nodes of the following graph is



a) MNOPQR
b) NQMPOR
c) QMNPRO

d) QMNPOR

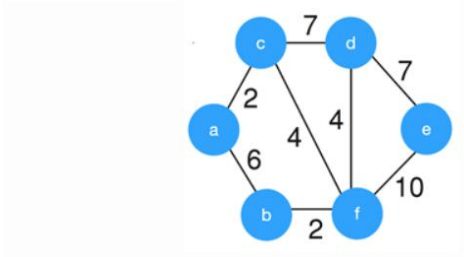
48. Consider the following graph,



Which are depth first traversals of the above graph?

a) I,II and IV
b) I and IV only
c) II,III and IV only
d) I,III and IV only

49. Consider the given graph.



What is the weight of the minimum spanning tree using the Kruskal's algorithm?

a) 24
b) 23
c) 15
d) 19

50. Which of the following is false about the Kruskal's algorithm?

a) It is a greedy algorithm
b) It constructs MST by selecting edges in increasing order of their weights
c) It can accept cycles in the MST
d) It uses union-find data structure

Part B

1 Discuss the components of Greedy Algorithm.

2 Compare Greedy technique with dynamic programming and divide and compare.

3. Draw the Characteristics of a good software design

4 What is brute force algorithm? List the strength and weakness of brute force algorithm.

5 Give the general plan for divide-and-conquer algorithms.

6 What is the general divide-and-conquer recurrence relation?

7 List out Disadvantages of Divide and Conquer Algorithm

8 Define dynamic programming and its features

9 Write the difference between the Greedy method and Dynamic programming.

10 What are the steps required to develop a greedy algorithm?

11 What are the labels in Prim's algorithm used for?

12 What is minimum spanning tree.

13 How are the vertices not in the tree split into?

14 What are the operations to be done after identifying a vertex u* to be added to the tree?

15 Explain Kruskal's algorithm of greedy method?

16 Explain the sum of subsets and with a suitable example?

17 Write backtracking knapsack Algorithm.

18 Compare brute force and Huffman method of greedy.

19 Write about Longest Common Subsequence using dynamic programming.

20 Explain about OBST with an example.

Part C

1. Explain in detail about greedy knapsack problem. Find an optimal solution to the knapsack instance n=7,m=15,$(P_1,P_2,P_3,P_4,P_5,P_6,P_7)$=(10,5,15,7,6,18,3) and $(W_1,W_2,W_3,W_4,W_5,W_6,W_7)$= (2,3,5,7,1,4,1)

2. Write dynamic programming solution for the travelling salesperson problem for the network with the cost adjacency matrix

$$\begin{pmatrix} 0 & 10 & 15 & 20 \\ 5 & 0 & 9 & 10 \\ 6 & 13 & 0 & 12 \\ 8 & 8 & 9 & 0 \end{pmatrix}$$

3. Explain in detail about Huffman code algorithm. Let A={a/5,d/5,c/12,d/13,e/16,f/45} be the letters and its frequency distribution in a text file. Compute a suitable Huffman coding to compress the data effectively and also compute optimal cost.

4. Write an algorithm to determine the sum of subsets for a given sum and a set of numbers. Draw the tree representation to solve the subset sum problem given the number set as {5,10,15,20,25} with the sum=30. Draw all the subsets.

5. Consider the travelling salesman instance defined by the cost matrix

$$\begin{bmatrix} \infty & 20 & 30 & 10 & 11 \\ 15 & \infty & 16 & 4 & 2 \\ 3 & 5 & \infty & 2 & 4 \\ 19 & 6 & 18 & \infty & 3 \\ 16 & 4 & 7 & 16 & \infty \end{bmatrix}$$

Find the optimal cost using branch and bound technique

6. Explain Divide And Conquer Method
7. Explain in detail about knapsack problem.
8. Explain Kruskal's Algorithm and Prim’s Algorithm
9. Explain Memory Function algorithm for the Knapsack problem
10. Explain in detail about Huffman tree.

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY, RAMAPURAM

CAMPUS COMPUTER SCIENCE AND ENGINEERING

QUESTION BANK

18CSC204J DESIGN AND ANALYSIS OF ALGORITMS

UNIT 4

Introduction to backtracking - branch and bound, N queen's problem – backtracking, Sum of subsets using backtracking, Complexity calculation of sum of subsets, Graph introduction, Hamiltonian circuit – backtracking, Branch and bound - Knapsack problem, Example and complexity calculation. Differentiate with dynamic and greedy, Travelling salesman problem using branch and bound, Travelling salesman problem using branch and bound example, Travelling salesman problem using branch and bound example, Time complexity calculation with an example, Graph algorithms, Depth first search and Breadth first search, Shortest path introduction, Floyd-Warshall Introduction, Floyd-Warshall with sample graph, Floyd-Warshall complexity

PART A

1. Which of the following is not a backtracking algorithm?

(A) Knight tour problem
(B) N queen problem
(C) Tower of hanoi
(D) M coloring problem
Answer: - C

2. Backtracking algorithm is implemented by constructing a tree of choices called as?

A) State-space tree
B) State-chart tree
C) Node tree
D) Backtracking tree
Answer: - A

3. What happens when the backtracking algorithm reaches a complete solution?

A) It backtracks to the root

B) It continues searching for other possible solutions
C) It traverses from a different route
D) Recursively traverses through the same route
Answer: - B

4. In what manner is a state-space tree for a backtracking algorithm constructed?
A**)** Depth-first search
B) Breadth-first search
C) Twice around the tree
D) Nearest neighbour first
Answer: - A

5. In general, backtracking can be used to solve?
A) Numerical problems
B) Exhaustive search
C) Combinatorial problems
D) Graph coloring problems
Answer: - C

6. Which one of the following is an application of the backtracking algorithm?
A) Finding the shortest path
B) Finding the efficient quantity to shop
C) Ludo
D) Crossword
Answer: - D

7. Who coined the term 'backtracking'?
A) Lehmer
B) Donald
C) Ross
D) Ford
Answer: - A

8. The problem of finding a subset of positive integers whose sum is equal to a given positive integer is called as?
A) n- queen problem
B) subset sum problem
C) knapsack problem
D) hamiltonian circuit problem
Answer: - B

9. The problem of placing n queens in a chessboard such that no two queens attack each other is called as?
A) n-queen problem
B) eight queens puzzle
C) four queens puzzle

D) 1-queen problem
Answer: - A

10. **In how many directions do queens attack each other?**
    A) 1
    B) 2
    C) 3
    D) 4
    Answer: - C

11. **Placing n-queens so that no two queens attack each other is called?** A) n-queen's problem
    B) 8-queen's problem
    C) Hamiltonian circuit problem
    D) subset sum problem
    Answer: - A

12. **Where is the n-queens problem implemented?**
    A) carom
    B) chess
    C) ludo
    D) cards
    Answer: - B

13. **Not more than 2 queens can occur in an n-queens problem.**
    A) true
    B) false
    Answer: - B

14. **In n-queen problem, how many values of n does not provide an optimal solution?** A) 1
    B) 2
    C) 3
    D) 4
    Answer: - B

15. **Which of the following methods can be used to solve n-queen's problem?** A) greedy algorithm
    B) divide and conquer
    C) iterative improvement
    D) backtracking
    Answer: - D

16. **Of the following given options, which one of the following is a correct option that provides an optimal solution for 4-queens problem?**

A) (3,1,4,2)
B) (2,3,1,4)
C) (4,3,2,1)
D) (4,2,3,1).
Answer: - A

17. **How many possible solutions exist for an 8-queen problem?**
A) 100
B) 98
C) 92
D) 88
Answer: - C

18. **How many possible solutions occur for a 10-queen problem?**
A) 850
B) 742
C) 842
D) 724.
Answer: - D

19. **The Knapsack problem is an example of _____**
A) Greedy algorithm
B) 2D dynamic programming
C) 1D dynamic programming
D) Divide and conquer
Answer: - B

20. **Which of the following methods can be used to solve the Knapsack problem?** A) Brute force algorithm
B) Recursion
C) Dynamic programming
D) Brute force, Recursion and Dynamic Programming
Answer: - D

21. **You are given a knapsack that can carry a maximum weight of 60. There are 4 items with weights {20, 30, 40, 70} and values {70, 80, 90, 200}. What is the maximum value of the items you can carry using the knapsack?**
A) 160
B) 200
C) 170
D) 90
Answer: - A

22. **Which of the following problems is equivalent to the 0-1 Knapsack problem?** A) You are given a bag that can carry a maximum weight of W. You are given N items which have a weight of {w1, w2, w3,…., wn} and a value of {v1, v2, v3,…., vn}. You

can break the items into smaller pieces. Choose the items in such a way that you get the maximum value
B) You are studying for an exam and you have to study N questions. The questions take {t1, t2, t3,…., tn} time(in hours) and carry {m1, m2, m3,…., mn} marks. You can study for a maximum of T hours. You can either study a question or leave it. Choose the questions in such a way that your score is maximized
C) You are given infinite coins of denominations {v1, v2, v3,….., vn} and a sum S. You have to find the minimum number of coins required to get the sum S
D) You are given a suitcase that can carry a maximum weight of 15kg. You are given 4 items which have a weight of {10, 20, 15,40} and a value of {1, 2, 3,4}. You can break the items into smaller pieces. Choose the items in such a way that you get the maximum value
Answer: - B

23. **What is the time complexity of the brute force algorithm used to solve the Knapsack problem?**
A) O(n)
B) O(n!)
C) $O(2^n)$
D) $O(n^3)$
Answer: - C

24. **Which of the following is/are property/properties of a dynamic programming problem?**
A) Optimal substructure
B) Overlapping subproblems
C) Greedy approach
D) Both optimal substructure and overlapping subproblems
Answer: - D

25. **If an optimal solution can be created for a problem by constructing optimal solutions for its subproblems, the problem possesses _____ property.** A) Overlapping subproblems
B) Optimal substructure
C) Memoization
D) Greedy
Answer: - B

26. **If a problem can be broken into subproblems which are reused several times, the problem possesses _____ property.**
A) Overlapping subproblems
B) Optimal substructure
C) Memoization
D) Greedy
Answer: - A

27. **If a problem can be solved by combining optimal solutions to non-overlapping problems, the strategy is called _____**

A) Dynamic programming
B) Greedy
C) Divide and conquer
D) Recursion
Answer: - C

28. **In dynamic programming, the technique of storing the previously calculated values is called _____**
A) Saving value property
B) Storing value property
C) Memoization
D) Mapping
Answer: - C

29. **When a top-down approach of dynamic programming is applied to a problem, it usually _____**
A) Decreases both, the time complexity and the space complexity
B) Decreases the time complexity and increases the space complexity
C) Increases the time complexity and decreases the space complexity
D) Increases both, the time complexity and the space complexity
Answer: - B

30. **Which of the following problems is NOT solved using dynamic programming?** A) 0/1 knapsack problem
B) Matrix chain multiplication problem
C) Edit distance problem
D) Fractional knapsack problem
Answer: - D

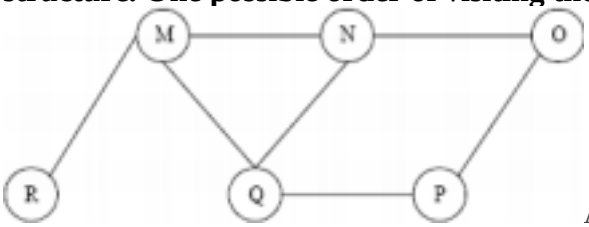
31. **Which of the following problems should be solved using dynamic programming?** A) Mergesort
B) Binary search
C) Longest common subsequence
D) Quicksort
Answer: - C

32. **Time Complexity of Breadth First Search is? (V - number of vertices, E - number of edges)**
A) O(V+E)
B) O(V)
C) O(E)
D) O(VE)
Answer: - A

33. **The spanning tree of connected graph with 10 vertices contains .....** A) 9 edges
B) 11 edges

C) 10 edges
D) 8 edges
Answer: - A

34. The Breadth First Search algorithm has been implemented using the queue data structure. One possible order of visiting the nodes of the following graph is



A) MNOPQR
B) NQMPOR
C) QMNPRO
D) QMNPOR
Answer: - C

35. What is the maximum height of queue (To keep track of un-explored nodes) required to process a connected Graph G1 which contains 'N' node using BFS algorithm? A) (N/2)-1
B) (N/2)/2
C) N-1
D) N
Answer: - C

PART B

1. What is meant by knapsack problem?
2. Define fractional knapsack problem.
3. Write the running time of 0/1 knapsack problem.
4. Write recurrence relation for 0/1 knapsack problem
5. What is meant by travelling salesperson problem?
6. What is the running time of dynamic programming TSP?
7. State if backtracking always produces optimal solution.
8. Define backtracking.
9. What are the two types of constraints used in backtracking?
10. What is meant by optimization problem?
11. Define Hamiltonian circuit problem.
12. What is Hamiltonian cycle in an undirected graph?
13. Define 8queens problem. 8. List out the application of backtracking.
14. Define promising node and non-promising node.
15. Give the explicit and implicit constraint for 8-queen problem.
16. How can we represent the solution for 8-queen problem?
17. Give the categories of the problem in backtracking.

18. Differentiate backtracking and over exhaustive search.
19. Find optimal solution for the knapsack instance n =3,w=[20,15,15],P =[40,25,25]and C =30
20. What is travelling salesperson problem?
21. What is the formula used to find upper bound for knapsack problem?
22. Differentiate between back tracking and branch and bound.
23. List out the application of branch and bound technique.
24. Analyze the time complexity for Warshall's and Floyd's algorithm. 25. Test the 0/1 knapsack problem.
26. Summarize Warshall's algorithm
27. Compare feasible and optimal solution.
28. Differentiate between DFS and BFS.
29. What is efficiency of DFS based algorithm for topological sorting
30. What are the different applications of DFS and BFS?

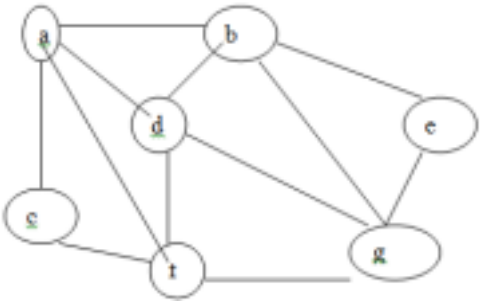
PART C

1. Describe the travelling salesman problem and discuss how to solve it using dynamic programming?
2. Find the optimal solution for the given knapsack problem

Item	1	2	3	4
weight	2	1	3	2
Value	$12	$10	$20	$15

.

3. Apply backtracking technique to solve the following instance of the subset sum problem S = [1,3,4,5} and d=11 16
4. Explain subset-sum problem and discuss the possible solution strategies using backtracking.
5. Explain N-queens problem with an algorithm.
6. Explain why backtracking is defined as a default procedure of last resort for solving problems.
7. Explain the subset-sum problem in detail by justifying it using backtracking algorithm.
8. Apply backtracking to the problem of finding a Hamiltonian circuit for the following graph.



9. What is backtracking?Explain in detail.
10. Solve the following instance of the knapsack problem by the branch and bound algorithm.

Item	Weight	Value
1	4	$40
2	7	#42
3	5	$25
4	3	$12
The Knapsack's capacity W=10		

11. Discuss the solution for knapsack problem using branch and bound technique. 12. What is branch and bound technique?Explain how knapsack problem could be solved using branch and bound technique.Solve the following instance of the knapsack problem by branch and bound algorithm for W=16

Item	Weight	Value in Rs.
1	10	100
2	7	63
3	8	56
4	4	12

13. What is branch and bound? Explain in detail.
14. Consider the below matrix for assignment problem involving persons and jobs.Explain in detail how branch and bound technique is useful in solving assignment problems.

	Job1	Job2	Job3	Job4
A	9	2	7	8
B	6	4	3	7
C	5	8	1	8
D	7	6	9	4

15. Discuss Warshall's algorithm with suitable example.

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY, RAMAPURAM CAMPUS

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

QUESTION BANK

18CSC204J DESIGN AND ANALYSIS OF ALGORITMS

UNIT 5

Introduction to randomization and approximation algorithm - Randomized hiring problem - Randomized quick sort, Complexity analysis - String matching algorithm, Examples - Rabin Karp algorithm for string matching, Example discussion - Approximation algorithm, Vertex covering - Introduction Complexity classes, P type problems - Introduction to NP type problems, Hamiltonian cycle problem - NP complete problem introduction, Satisfiability problem - NP hard problems - Examples

PART A

1. What is a Rabin and Karp Algorithm?

(A) String Matching Algorithm
(B) Shortest Path Algorithm
(C) Minimum spanning tree Algorithm
(D) Approximation Algorithm
Answer: - A

2. What is the pre-processing time of Rabin and Karp Algorithm?

A) Theta(m^2)
B) Theta(mlogn)
C) Theta(m)
D) Big-Oh(n)
Answer: - C

3. Rabin Karp Algorithm makes use of elementary number theoretic notions.

A) True
B) FALSE
Answer: - A

4. Given a pattern of length- 5 window, find the spurious hit in the given text string.

Pattern: 3 1 4 1 5
Modulus: 13
Index: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
Text: 2 3 5 9 0 2 3 1 4 1 5 2 6 7 3 9 9 2 1 3 9

A**)** 6-10
B) 12-16
C) 3-7
D) 13-17
Answer: - D

5. What is the basic principle in Rabin Karp algorithm?

A) Hashing
B) Sorting
C) Augmenting
D) Dynamic Programming
Answer: - A

6. The worst-case efficiency of solving a problem in polynomial time is?

A)O(p(n))
B)O(p(nlogn))
C) $O(p(n^2))$
D) O(p(m log n))
Answer: - A

7. Problems that can be solved in polynomial time are known as?

A) Intractable
B) Tractable
C) Decision
D) Complete
Answer: - B

8. _____ is the class of decision problems that can be solved by non-deterministic polynomial algorithms.

A) NP
B) P
C) Hard
D) Complete
Answer: - A

9. The Euler's circuit problem can be solved in?

A) O(N)
B) O(N log N)
C)O(logN)
D) $O(N^2)$
Answer: - D

10. To which of the following class does a CNF-satisfiability problem belong?

A) NP class
B) P class
C) NP complete
D) NP hard
Answer: - C

11. Quick sort uses which of the following algorithm to implement sorting?

A) backtracking
B) greedy algorithm
C) divide and conquer
D) dynamic programming
Answer: - C

12. What is the worst case time complexity of randomized quicksort?

A) O(n)
B) O(n log n)
C)$O(n^2)$
D) $O(n^2 \log n)$
Answer: - C

13. What is the purpose of using randomized quick sort over standard quick sort?

A) so as to avoid worst case time complexity
B) so as to avoid worst case space complexity
C) to improve accuracy of output
D) to improve average case time complexity
Answer: - A

14. Which of the following is incorrect about randomized quicksort?

A) it has the same time complexity as standard quick sort
B) it has the same space complexity as standard quick sort
C) it is an in-place sorting algorithm
D) it cannot have a time complexity of O(n2) in any case.
Answer: - D

15. Which of the following is the fastest algorithm in string matching field?

A) Boyer-Moore's algorithm
B) String matching algorithm
C) Quick search algorithm
D) Linear search algorithm
Answer: - C

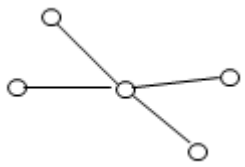
16. What is vertex coloring of a graph?
A) A condition where any two vertices having a common edge should not have same color
B) A condition where any two vertices having a common edge should always have same color
C) A condition where all vertices should have a different color
D) A condition where all vertices should have same color
Answer: - A

17. How many edges will a tree consisting of N nodes have?
A) Log(N)
B) N
C) N-1
D) N+1
Answer: - C

18. Minimum number of unique colors required for vertex coloring of a graph is called?
A) vertex matching
B) chromatic index
C) chromatic number
D) color number.
Answer: - C

19. How many unique colors will be required for proper vertex coloring of an empty graph having n vertices?
A) 0
B) 1
C) n
D) n!
Answer: - C

20. What will be the chromatic number of the following graph?



A) 1
B) 2
C) 3

D) 4
Answer: - B

21. Assuming P != NP, which of the following is true ?
A) NP-complete = NP
B) NP-complete \cap P = \Phi
C) NP-hard = NP
D) P = NP-complete
Answer: - B

22. Let X be a problem that belongs to the class NP. Then which one of the following is TRUE?
A) There is no polynomial time algorithm for X.
B) If X can be solved deterministically in polynomial time, then P = NP.
C) If X is NP-hard, then it is NP-complete.
D) X may be undecidable.
NP Complete
Answer: - C

23. Which of the following statements are TRUE?
1. The problem of determining whether there exists a cycle in an undirected graph is in P.
2. The problem of determining whether there exists a cycle in an undirected graph is in NP.
3. If a problem A is NP-Complete, there exists a non-deterministic polynomial time algorithm to solve A.
A) 1, 2 and 3
B) 1 and 2 only
C) 2 and 3 only
D) 1 and 3 only
Answer: - A

24. Consider the following two problems on undirected graphs
α : Given G(V, E), does G have an independent set of size | V | - 4?
β : Given G(V, E), does G have an independent set of size 5?
Which one of the following is TRUE?

A) α is in P and β is NP-complete
B) α is NP-complete and β is in P
C) Both α and β are NP-complete
D) Both α and β are in P
Answer: - C

25. Which of the following algorithm can be used to solve the Hamiltonian path problem efficiently?
A) branch and bound

B) iterative improvement
C) divide and conquer
D) greedy algorithm
Answer: - A

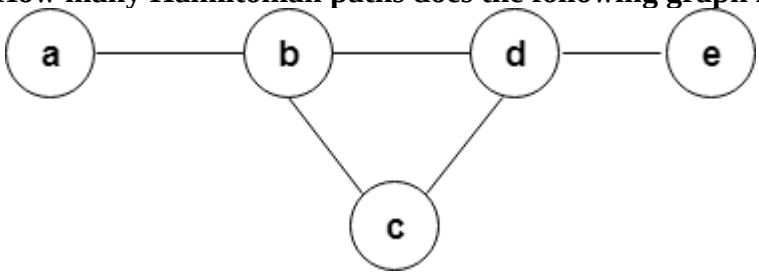
26. Hamiltonian path problem is _____
A) NP problem
B) N class problem
C) P class problem
D) NP complete problem
Answer: - D

27. There is no existing relationship between a Hamiltonian path problem and Hamiltonian circuit problem.
A) true
B) false
Answer: - B

28. Which of the following problems is similar to that of a Hamiltonian path problem?
A) knapsack problem
B) closest pair problem
C) travelling salesman problem
D) assignment problem
Answer: - C

29. In what time can the Hamiltonian path problem can be solved using dynamic programming?
A) O(N)
B) O(N log N)
C) O(N2)
D) O(N2 2N)
Answer: - D

30. How many Hamiltonian paths does the following graph have?



A) 1
B) 2
C) 3
D) 4
Answer: - A

31. A node is said to be _____ if it has a possibility of reaching a complete solution.
A) Non-promising
B) Promising
C) Succeeding
D) Preceding
Answer: - B

32. Minimum number of unique colors required for vertex coloring of a graph is called?
A) vertex matching
B) chromatic index
C) chromatic number
D) color number
Answer: C

PART B

1. Define NP hard and NP completeness.
2. Compare NP hard and NP completeness.
3. Write Short notes on “the class P and NP problem”.
4. How NP Hard problems are different from NP Complete?
5. Whether class P solves a problem in polynomial time? Justify.
6. An NP hard problem can be solved in deterministic polynomial time, how?
7. Give examples for NP Complete problems
8. State the property of NP complete problem.
9. Define adversary method.
10. Define lower bound.
11. What type of output yields trivial lower bound?
12. What is information theoretic lower bound?
13. Define complexity theory.
14. What is halting problem?

15. What is CNFs satisfiablity problem?

16. Define Matching.

17. Define a bipartite graph.

18. How will you check the stability?

19. What is stable marriage problem?

20. Define the term stable pair

21. What do you mean by perfect match in bipartite graph?

22. Write Rabin Karp string matching algorithm

23. Describe Hamiltonian cycle problem

PART C

1. Describe in detail about P and NP Problems
2. Write short notes on NP Complete Problem
3. Write short notes on the following using approximation Algorithm

   i) Nearest –neighbor algorithm with example

   ii) Multi fragment heuristic algorithm with example
4. Describe in detail about Twice around the tree algorithm with example
5. Explain local search heuristic with example
6. Explain Approximation Algorithms for the Travelling Salesman Problem
7. Explain the Assignment problem in Branch and bound with Example.
8. Suggest an approximation algorithm for TSP. Assume that the cost function satisfies the triangle inequality.
9. Using an example prove that, satisfiability of Boolean formula in 3- Conjunctive Normal Form is NP – complete.
10. State the relationships among the complexity class algorithms with the help of

neat diagrams

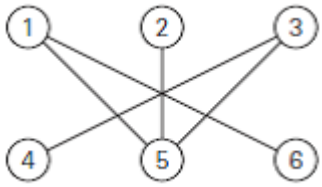
11. Explain the algorithm for stable marriage problem and prove the theorem with Example

12. Consider an instance of the stable marriage problem given by the ranking matrix

	A	**B**	**C**
α	1,3	2,2	3,1
β	3,1	1,3	2,2
γ	2,2	3,1	1,3

For each of its marriage matching's, indicate whether it is stable or not

13. Apply the maximum matching algorithm to the following bipartite graphs



14. Write the algorithm for maximum matching in Bipartite Graphs and prove the theorem With example

15. Explain the algorithm:
    i. Blocking pair
    ii. Stable marriage problem
    iii. Man optimal
    iv. Women optimal

16. Explain briefly on minimum weight perfect matching algorithm.

17. Explain briefly on reducing bipartite graph to net flow

18. Explain local search heuristic with example

19. Consider the following minimization problem:
    DEGREE BOUNDED SPANNING TREE:
    Instance: Graph G = (V,E)
    Solution:: A spanning tree T of G
    Value: Maximum degree of T

Goal: Find a solution with minimum value.

20. Consider the following scheduling problem. You are given n jobs where job i is specied by an earliest start time si and a processing time pi. In homework 1, we considered a preemptive version of this problem and gave a greedy algorithm to give an optimal preemptive schedule. In this problem we consider the non-preemptive version of this scheduling problem. Here a job CANNOT be suspended but rather must be performed in a contiguous time interval. Consider the following heuristic for the non-preemptive problem: schedule the jobs in the order in which they complete in an optimal preemptive schedule starting each job as soon as the one before it completes. You are to prove that this algorithm is a 2-approximation algorithm.

21. Using Rabin karp string matching algorithm match the given pattern P with given string S.
    P = 745
    S = 745727457

22. Using KMP string matching algorithm, find the occurrence of the given patter P in the given text T.
    T ß ABABACAB
    P ß ABAB

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.

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?
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.

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.

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.

UNIT II: DIVIDE AND CONQUER

Introduction-Divide and Conquer, Maximum Sub array Problem- Binary Search, Complexity of binary search- Merge sort, Time complexity analysis-Quick sort and its Time complexity analysis, Best case, Worst case, Average case analysis - Strassen's Matrix multiplication and its recurrence relation, Time complexity analysis of Merge sort - Largest sub-array sum, Time complexity analysis of Largest sub-array sum- Master Theorem Proof, Master theorem examples- Finding Maximum and Minimum in an array, Time complexity analysis- Examples- Algorithm for finding closest pair problem, Convex Hull problem

PART-A

1.) Partition and exchange sort is_____

A. quick sort

B. tree sort

C. heap sort

D. bubble sort

ANSWER: A

2) Which of the following is not the required condition for binary search algorithm?

A. The list must be sorted

B. There should be the direct access to the middle element in any sub list

C. There must be mechanism to delete and/or insert elements in list.

D. Number values should only be present

ANSWER: C

3) Which of the following sorting algorithm is of divide and conquer type?

A. Bubble sort

B. Insertion sort

C. Merge sort

D. Selection sort

ANSWER: C

4) _____order is the best possible for array sorting algorithm which sorts n item.

A. O (n logn)

B. O (n2)

C. O (n+logn)

D. O (logn)

ANSWER: C

5) The complexity of merge sort algorithm is _____

A. O (n)

B. O (logn)

C. O (n2)

D. O (n logn)

ANSWER: D

6) Binary search algorithm cannot be applied to _____

A. sorted linked list

B. sorted binary trees

C. sorted linear array

D. pointer array

ANSWER: A

7) Which of the following is not a limitation of binary search algorithm?

A. must use a sorted array

B. requirement of sorted array is expensive when a lot of insertion and deletions are needed

C. there must be a mechanism to access middle element directly

D. binary search algorithm is not efficient when the data elements more than 1500.

ANSWER: D

8) Which of the following is an external sorting?

A. Insertion Sort

B. Bubble Sort

C. Merge Sort

D. Tree Sort

ANSWER: B

9) Merging k sorted tables into a single sorted table is called _____

A. k way merging

B. k th merge

C. k+1 merge

D. k-1 merge

ANSWER: A

10) The operation that combines the element is of A and B in a single sorted list C with n=r+s element is called _____

A. Inserting

B. Mixing

C. Merging

D. Sharing

ANSWER: C

11) Which of the following is a stable sorting algorithm?

a) Merge sort

b) typical in-place quick sort

c) Heap sort

d) Selection sort

ANSWER: A

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

a) Selection sort

b) Heap sort

c) Quick sort

d) Merge sort

ANSWER: D

13)The time complexity of a quick sort algorithm which makes use of median, found by an O(n) algorithm, as pivot element is

a) $O(n^2)$

b) O(nlogn)

c) O(nloglogn)

d) O(n)

ANSWER: B

14) 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

ANSWER: C

15) Merge sort uses

a) Divide-and-conquer

b) Backtracking

c) Heuristic approach

d) Greedy approach

ANSWER: A

16)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)

ANSWER: C

17) The running time of Strassen's algorithm for matrix multiplication is

(A) Θ (n) (B) Θ (n^3) **(**C) Θ (n^2) (D) Θ ($n^{2.81}$)

ANSWER: D

18) The Stassen's algorithm's time complexity is

(A) 0(n) (B) 0(n^2) (C) 0(n^2.80) (D) 0(n^2.81)

ANSWER: C

19) Which algorithm is used for matrix multiplication?

a. Simple algorithm

b. Specific algorithm

c. Strassen algorithm

d. Addition algorithm

ANSWER: C

20) Which algorithm is a divided and conquer algorithm that is asymptotically faster:

a. Simple algorithm

b. Specific algorithm

c. Strassen algorithm

d. Addition algorithm

ANSWER: C

21) Which algorithm is named after Volker Strassen

a. Strassen algorithm

b. Matrix algorithm

c. Both

d. None of these

ANSWER: A

22) Which of the following algorithms is NOT a divide & conquer algorithm by nature?

(A) Euclidean algorithm to compute the greatest common divisor

(B) Heap Sort

(C) Closest pair problem

(D) Quick Sort

Answer: B

23). what is the average case time complexity of merge sort?

a) O (n log n)

b) O (n2)

c) O (n2 log n)

d) O (n log n2)

ANSWER: A

24). which of the following method is used for sorting in merge sort?

a) Merging

b) Partitioning

c) Selection

d) Exchanging

ANSWER: A

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

a) Quick sort

b) Cocktail sort

c) Bubble sort

d) Merge sort

ANSWER: A

26) What is the runtime efficiency of using brute force technique for the closest pair problem?

a) O (N)

b) O (N log N)

c) O (N2)

d) O (N3 log N)

ANSWER: C

27) What is the basic operation of closest pair algorithm using brute force technique?

a) Euclidean distance

b) Radius

c) Area

d) Manhattan distance

ANSWER: A

28) _____ is a method of constructing a smallest polygon out of n given points.

a) Closest pair problem

b) Quick hull problem

c) Path compression

d) union-by-rank

ANSWER: B

29) Find the maximum sub-array sum for the given elements.

{-2, -1, -3, -4, -1, -2, -1, -5, -4}

a) -3

b) 5

c) 3

d) -1

ANSWER: D

30) Master's theorem is used for?

a) Solving recurrences

b) Solving iterative relations

c) Analyzing loops

d) Calculating the time complexity of any code

ANSWER: A

PART-B

1. How the large integers are multiplies using divide and conquer technique?
2. Give the recurrence relation for divide and conquer.
3. Define Master Theorem.
4. Find the order of growth for the following recurrence.
5. T (n) =4T(n/2)+n2, T(1)=1
6. Give the examples for divide and conquer method.
7. Write the control abstraction for divide and conquer technique.
8. What are the best case, worst case and average case complexity of Quick sort?
9. Solve the average case recurrence for quick sort.
10. How to search an element using binary search?
11. What are the merits of binary search?
12. What are the merits of divide and conquer technique?
13. What are the demerits of binary search?
14. Trace the operation of the binary search algorithm for the input –15, –6, 0, 7, 9, 23, 54, 82, 101, 112, 125, 131, 142, 151, if you are searching for the element 9.
15. What is the time complexity of binary search?
16. What is average case efficiency and worst case complexity of binary search?
17. Derive the complexity of Binary Search algorithm.
18. What are the best case, worst case and average case complexity of binary Search?
19. Prove the equality $a \log_b c = c \log_b a$.
20. Solve the average case recurrence for quick sort.
21. How the operations performed in Strassen's Matrix multiplication?
22. Compute 2101*1130 by applying the divide and conquer algorithm.
23. What is the time complexity of closest pair and quick hull problem?

PART C

1) Determine the efficiency of divide and conquer algorithms.

2) Explain and analyze the merge sort algorithm.

3) How quick sort can be improved?

4) Explain the binary searching algorithm in detail, with an example. Show the worst case efficiency of binary search is in Θ (log n)

5) Explain the divide and conquer strategy with examples.

6) Apply quick sort to sort the list E, X, A, M, P, L, E in alphabetical order. Draw the tree of the recursive calls made.

   i. Write the best case input for the quick sort

   ii. Find the best case time efficiency for the quick sort

iii. Are the arrays made up of all equal elements the worst case input, the best case input or neither?

7) Write a pseudo code for divide & conquer algorithm for finding the position of the largest element in an array of numbers.

8) Write a quick sort algorithm and derive the worst case and average case complexity class of this algorithm.

9) Discuss the efficiency of quick sort algorithm.

10) Explain how the merge sort can be viewed as a recursive application of the Divide and conquer methodology. Suggest a pseudo code for merge sort and analyze its complexities. Trace its application to the following data set 9, 4, 3,8,6,2,1,5,7.

11) Explain Strassen's matrix multiplication. Evaluate its efficiency.

12) Discuss about the Quick Hull Algorithm.

13) Explain in Detail about the Closest pair Algorithm

14) Explain in detail about min-max problem using divide and conquer and derive its time complexity.

15) Discuss the advantages and disadvantages of Divide and Conquer Algorithm?

UNIT III

Introduction-Greedy and Dynamic Programming, Examples of problems that can be solved by using greedy and dynamic approach Huffman coding using greedy approach, Comparison of brute force and Huffman method of encoding Knapsack problem using greedy approach, Complexity derivation of knapsack using greedy Tree traversals, Minimum spanning tree - Greedy, Kruskal's algorithm - greedy Minimum spanning tree - Prim's algorithm, Introduction to dynamic programming 0/1 knapsack problem, Complexity calculation of knapsack problem Matrix chain multiplication using dynamic programming, Complexity of matrix chain

multiplication Longest common subsequence using dynamic programming, Explanation of LCS with an example Optimal binary search tree (OBST)using dynamic programming, Explanation of OBST with an example

PART A

1. ---------- is a Boolean-valued function that determines whether x can be included into the solution vector

   a) Overlapping subproblems

   b) Fleasible solution

   c) Memoization

   d) Greedy

   Answer **B**

2. Trees with edge with weights are called ----------

   a) weighted tree

   b) unweighted tree

   c) bruteforce

   d) Greedy

   Answer **A**

3. ---------- is to determine an optimal placement of booster

   a) Weighted tree

   b) Vertex

   c) Tree Vertex Splitting Problem (TVSP)

   d) Greedy

   Answer **C**

4. The order in which TVS visits that computes the delay values of the nodes of the tree is called the----------.

a)treeorder

b) inorder

c) preorder

d) postorder

Answer **D**

5. Algorithm TVS takes ---------- time, where n is the number f nodes in the tree

a)O(N)

b)Ω(n log n)

c)O (n2log n)

d)O(n log n)

Answer **A**

6. ---------- is a greedy method to obtain a minimum-cost spanning tree builds this tree edge by edge

a)Prim's algorithm

b)Dynamic algorithm

c)Greedy algorithm

d)Dynamic algorithm

Answer **A**

7. Kruskal's algorithm (choose best non-cycle edge) is better than Prim's Choose best tree edge) when the graph has relatively few edges

a)True

b)False

Answer **A**

8. Two sorted files containing n and m records respectively could be merged together to obtain one sorted file in time ----------.

a) Ω(n log n)

b) **O(n+m)**

c) O (n2log n)

d)O(n log n)

Answer **B**

9. The two-way merge pattern scan be represented by----------

a) Weighted tree

b) Vertex

c) Binary merge tree

d) Greedy

Answer **C**

10. What algorithm technique is used in the implementation of Kruskal solution for the MST?

a) greedy technique

b) divide-and-conquer technique

c) dynamic programming technique

d) the algorithm combines more than one of the above techniques

Answer **A**

11. The function Tree of Algorithm uses the ------- stated to obtain a two-way merge tree for n file

a) divide-and-conquer technique

b) greedy rule

c) dynamic programming technique

d) the algorithm combines more than one of the above techniques

Answer **B**

12. A decode tree is a---------- in which external nodes represent messages.

a) minimum spanning tree

b) B tree

c) Binary tree

d) AVL tree

Answer **B**

13. The ----------in the code word for a message determine the branching needed at each level of the decode tree to reach the correct external node.

a) binary bits

b) decoder

c) encoder

d) binary bytes

Answer **A**

14. The cost of decoding a --------is proportional to the number of bits in the code

a) binary bits

b) code word

c) data

d) binary bytes

Answer **B**

15. What is the edges on the shortest paths from a vertex v to all remaining vertices in a connected undirected graph G form a spanning tree of G is called?

a) MST

b) shortest-path spanning tree

c) binary tree

d) AVL tree

Answer **B**

16. ----------is an algorithm design method that can be used when the

solution to a problem can be viewed as the result of a sequence of decisions.

a) Dynamic Programming

b) Greedy method

c) Huffman coding

d) Tree traversal

Answer **A**

17. What is an another important feature of the dynamic programming approach that optimal solutions are retained so as to avoid re-computing their values.

a) Dynamic Programming

b) Greedy method

c) Huffman coding

d) Tree traversal

Answer **A**

18. --------- often drastically reduces the amount of enumeration by avoiding the enumeration of some decision sequences that cannot possibly be optimal.

a) Dynamic Programming

b) Greedy method

c) Huffman coding

d) Tree traversal

Answer **A**

19. In the ----------only one decision sequence is ever generated.

a) Dynamic Programming

b) Greedy method

c) Huffman coding

d) Tree traversal

Answer **B**

20. Dynamic programming algorithms solve the---------- to obtain a solution to the given problem instance

a) optimistic

b) Greedy method

c) Huffman coding

d) recurrence

Answer **D**

21. A dynamic programming formulation for a k-stage graph problem is obtained by first noticing that every s to t path is the result of a sequence of ---------- decision.

a) k

b) k-1

c) k-2

d) 2k

Answer **C**

22. Which of the following problems is NOT solved using dynamic programming?

a) 0/1 knapsack problem

b) Matrix chain multiplication problem

c) Edit distance problem

d) Fractional knapsack problem

Answer **D**

23. The problem of ---------is to identify a minimum-cost sequence of edit operations that will transform X into Y.

a) 0/1 knapsack problem

b) Matrix chain multiplication problem

c) Edit distance problem

d) string editing

Answer **D**

24. In Knapsack problem, the best strategy to get the optimal solution, where Pi, Wi is the Profit, Weight associated with each of the Xith object respectively is to

a)Arrange the values Pi/Wi in ascending order

b)Arrange the values Pi/Xi in ascending order

c)Arrange the values Pi/Wiin descending order

d)Arrange the values Pi/Xiin descending order

Answer **D**

25. Greedy job scheduling with deadlines algorithms' complexity is defined as a)O(N)

b)Ω(n log n)

c)O (n2log n)

d)O(n log n)

Answer **B**

26. In Huffman coding, data in a tree always occur?

a) roots

b) leaves

c) left sub trees

d) right sub trees

Answer **B**

27. The multistage graph problem can also be solved using the ----------

a) backward approach

b) forward approach

c) brute force approach

d) right sub trees

Answer **B**

28. The all-pairs ----------problem is to determine a matrix A such that A(i,j)is the length of a shortest path from i to j.

a) backward approach

b) forward approach

c) brute force approach

d) shortest-path

Answer **D**

29. Which of the following methods can be used to solve the Knapsack problem?

a) Brute force algorithm

b) Recursion

c) Dynamic programming

d) Brute force, Recursion and Dynamic Programming

Answer **D**

30. The inorder and preorder traversal of a binary tree are d b e a f c g and a b d e c f g, respectively. The postorder traversal of the binary tree is:

a) d e b f g c a

b) e d b g f c a

c) e d b f g c a

d) d e f g b c a

Answer **A**

31. Which of the following pairs of traversals is not sufficient to build a binary tree from the given traversals?

a) Preorder and Inorder

b) Preorder and Postorder

c) Inorder and Postorder

d) Inorder and levelorder

Answer **D**

32. Consider the following C program segment

```
struct CellNode
{
  struct CelINode *leftchild;
  int element;
  struct CelINode *rightChild;
}
int Dosomething(struct CelINode *ptr)
{
   int value = 0;
   if (ptr != NULL)
   {
     if (ptr->leftChild != NULL)
       value = 1 + DoSomething(ptr->leftChild);
     if (ptr->rightChild != NULL)
       value = max(value, 1 + DoSomething(ptr->rightChild));
   }
   return (value);
}
```

The value returned by the function DoSomething when a pointer to the root of a non empty tree is passed as argument is

a) The number of leaf nodes in the tree

b) The number of nodes in the tree

c) The number of internal nodes in the tree

d) The height of the tree

Answer **D**

33. Given items as {value, weight} pairs {{60, 20}, {50, 25}, {20, 5}}. The capacity of knapsack=40. Find the maximum value output assuming items to be divisible and nondivisible respectively.

a) 100,80

b) 110,70

c) 130,110

d) 110,80

Answer **D**

34. Given items as {value, weight} pairs {{40, 20}, {30, 10}, {20, 5}}. The capacity of knapsack=20. Find the maximum value output assuming items to be divisible

a) 60

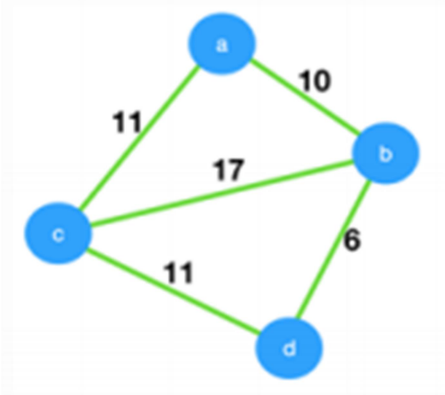
b) 80

c) 100

d) 40

Answer **A**

35. Consider the given graph.



What is the weight of the minimum spanning tree using the Prim's algorithm, starting from vertex a?

a) 23

b) 28

c) 27

d) 11

Answer **C**

36. Worst case is the worst case time complexity of Prim's algorithm if adjacency matrix is used?

a) O(log V)

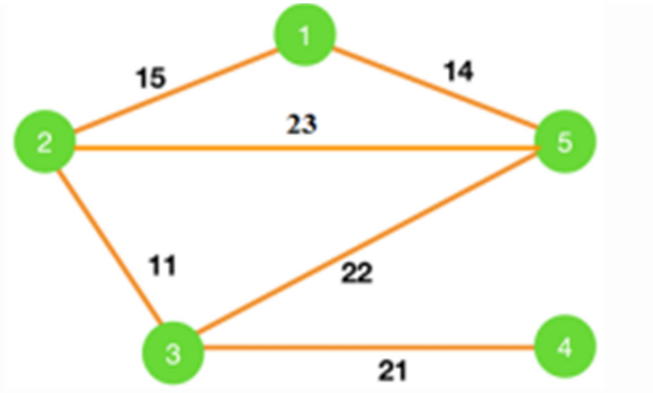
b) $O(V^2)$

c) $O(E^2)$

d) O(V log E)

Answer **B**

37. Consider the graph shown below.



Which of the following edges form the MST of the given graph using Prim'a algorithm, starting from vertex 4.

a) (4-3)(5-3)(2-3)(1-2)

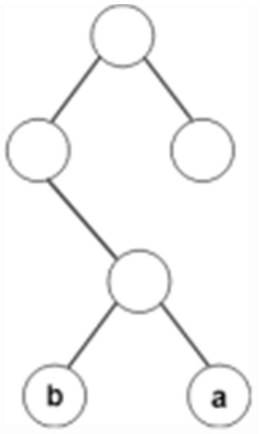
b) (4-3)(3-5)(5-1)(1-2)

c) (4-3)(3-5)(5-2)(1-5)

d) (4-3)(3-2)(2-1)(1-5)

Answer **D**

38. From the following given tree, what is the code word for the

character ‘a’? 

a) 011

b) 010

c) 100

d) 101

Answer **A**

39. What will be the cost of the code if character c_i is at depth d_i and occurs at frequency f_i?

a) $c_i f_i$

b) $\int c_i f_i$

c) $\sum f_i d_i$

d) $f_i d_i$

Answer **C**

40. What is the running time of the Huffman encoding algorithm?

a) O(C)

b) O(log C)

c) O(C log C)

d) O(N log C)

Answer **C**

41. The weighted array used in TVS problems for the following binary tree is _____

a) [1,2,3,0,0,4,0,5,6]

b) [1,2,3,0,0,4,0,5,0,0,0,0,6]

c) [1,2,3,4,5,6]

d) [1,2,3,0,0,4,5,6]

Answer **B**

42. What is the time complexity of the brute force algorithm used to find the longest common subsequence?

a) O(n)

b) $O(n^2)$

c) $O(n^3)$

d) $O(2^n)$

Answer **D**

43. Find the longest increasing subsequence for the given sequence:

{10, -10, 12, 9, 10, 15, 13, 14}

a) {10, 12, 15}

b) {10, 12, 13, 14}

c) {-10, 12, 13, 14}

d) {-10, 9, 10, 13, 14}

Answer **D**

44. What is the space complexity of the following dynamic programming implementation used to find the length of the longest increasing subsequence?

```
#include<stdio.h>
int longest_inc_sub(int *arr, int len)
{
 int i, j, tmp_max;
 int LIS[len]; // array to store the lengths of the longest increasing
subsequence   LIS[0]=1;
 for(i = 1; i < len; i++)
```

```
{
tmp_max = 0;
            for(j = 0; j < i; j++)
            {
            if(arr[j] < arr[i])
            {
                        if(LIS[j] > tmp_max)
                        tmp_max = LIS[j];
            }
}
            LIS[i] = tmp_max + 1;
}
 int max = LIS[0];
 for(i = 0; i < len; i++)
            if(LIS[i] > max)
             max = LIS[i];
  return max;
}
int main()
{
 int arr[] = {10,22,9,33,21,50,41,60,80}, len = 9;
 int ans = longest_inc_sub(arr, len);
 printf("%d",ans);
 return 0;
}
```

a) O(1)

b) O(n)

c) O(n2)

d) O(nlogn)

Answer **B**

45. The Breadth First Search algorithm has been implemented using the queue data structure. One possible order of visiting the nodes of the following

graph is

a) O(1)

b) O(n)

c) O(n2)

d) O(nlogn)

Answer **B**

46. Uniform-cost search expands the node n with the _____

a) Lowest path cost

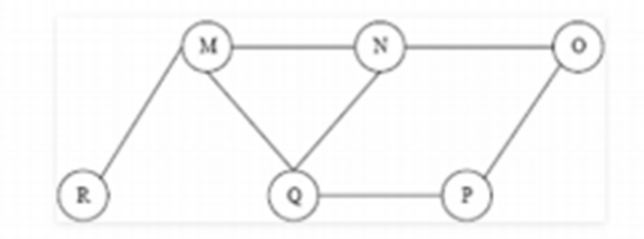
b) Heuristic cost

c) Highest path cost

d) Average path cost

Answer **A**

47. The Breadth First Search algorithm has been implemented using the queue data structure. One possible order of visiting the nodes of the following graph is



a) MNOPQR

b) NQMPOR

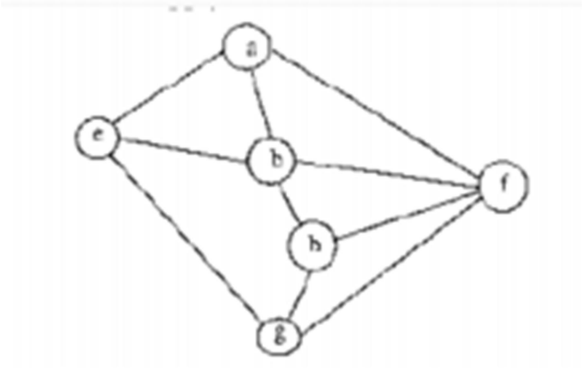
c) QMNPRO

d) QMNPOR

Answer **C**

48. Consider the following graph,

Which are depth first traversals of the above graph?



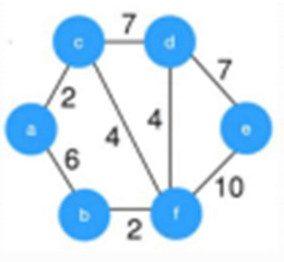
a) I,II and IV

b) I and IV only

c) II,III and IV only

d) I,III and IV only

49. Consider the given graph.



What is the weight of the minimum spanning tree using the Kruskal's algorithm?

a) 24

b) 23

c) 15

d) 19

Answer **D**

50. Which of the following is false about the Kruskal's algorithm?

a) It is a greedy algorithm

b) It constructs MST by selecting edges in increasing order of their weights

c) It can accept cycles in the MST

d) It uses union-find data structure

Answer **C**

PART B

1 Discuss the components of Greedy Algorithm.

2 Compare Greedy technique with dynamic programming and divide and compare.

3. Draw the Characteristics of a good software design

4 What is brute force algorithm? List the strength and weakness of brute force algorithm.

5 Give the general plan for divide-and-conquer algorithms.

6 What is the general divide-and-conquer recurrence relation?

7 List out Disadvantages of Divide and Conquer Algorithm

8 Define dynamic programming and its features

9 Write the difference between the Greedy method and Dynamic programming.

10 What are the steps required to develop a greedy algorithm?

11 What are the labels in Prim's algorithm used for?

12 What is minimum spanning tree.

13 How are the vertices not in the tree split into?

14 What are the operations to be done after identifying a vertex u* to be added to the tree?

15 Explain Kruskal's algorithm of greedy method?

16 Explain the sum of subsets and with a suitable example?

17 Write backtracking knapsack Algorithm.

18 Compare brute force and Huffman method of greedy.

19 Write about Longest Common Subsequence using dynamic programming.

20 Explain about OBST with an example.

PART C

1. Explain in detail about greedy knapsack problem. Find an optimal solution to the knapsack instance n=7,m=15,(P1,P2,P3,P4,P5,P6,P7)=(10,5,15,7,6,18,3) and (W1,W2,W3,W4,W5,W6,W7)= (2,3,5,7,1,4,1)
2. Write dynamic programming solution for the travelling salesperson problem for the network with the cost adjacency matrix

$$\begin{pmatrix} 0 & 10 & 15 & 20 \\ 5 & 0 & 9 & 10 \\ 6 & 13 & 0 & 12 \\ 8 & 8 & 9 & 0 \end{pmatrix}$$

3. Explain in detail about Huffman code algorithm. Let A={a/5,d/5,c/12,d/13,e/16,f/45} be the letters and its frequency distribution in a text file. Compute a suitable Huffman coding to compress the data effectively and also compute optimal cost.
4. Write an algorithm to determine the sum of subsets for a given sum and a set of numbers. Draw the tree representation to solve the subset sum problem given the number set as {5,10,15,20,25} with the sum=30. Draw all the subsets.
5. Consider the travelling salesman instance defined by the cost matrix

$$\begin{bmatrix} \infty & 20 & 30 & 10 & 11 \\ 15 & \infty & 16 & 4 & 2 \\ 3 & 5 & \infty & 2 & 4 \\ 19 & 6 & 18 & \infty & 3 \\ 16 & 4 & 7 & 16 & \infty \end{bmatrix}$$

   Find the optimal cost using branch and bound technique
6. Explain Divide And Conquer Method
7. Explain in detail about knapsack problem.
8. Explain Kruskal's Algorithm and Prim's Algorithm
9. Explain Memory Function algorithm for the Knapsack problem
10. Explain in detail about Huffman tree.

UNIT IV

Introduction to backtracking - branch and bound, N queen's problem – backtracking, Sum of subsets using backtracking, Complexity calculation of sum of subsets, Graph introduction, Hamiltonian circuit – backtracking, Branch and bound - Knapsack problem, Example and complexity calculation. Differentiate with dynamic and greedy, Travelling salesman problem using branch and bound, Travelling salesman problem using branch and bound example, Travelling salesman problem using branch and bound example, Time complexity calculation with an example, Graph algorithms, Depth first search and Breadth first search, Shortest path introduction, Floyd-Warshall Introduction, Floyd-Warshall with sample graph, Floyd-Warshall complexity

PART A

1. Which of the following is not a backtracking algorithm?

(A) Knight tour problem

(B) N queen problem

(C) Tower of hanoi

(D) M coloring problem

Answer: - C

2. Backtracking algorithm is implemented by constructing a tree of choices called as?

A) State-space tree

B) State-chart tree

C) Node tree

D) Backtracking tree

Answer: - A

3. What happens when the backtracking algorithm reaches a complete solution?

A) It backtracks to the root

B) It continues searching for other possible solutions

C) It traverses from a different route

D) Recursively traverses through the same route

Answer: - B

4. In what manner is a state-space tree for a backtracking algorithm constructed?

A**)** Depth-first search

B) Breadth-first search

C) Twice around the tree

D) Nearest neighbour first

Answer: - A

5. In general, backtracking can be used to solve?

A) Numerical problems

B) Exhaustive search

C) Combinatorial problems

D) Graph coloring problems

Answer: - C

6. Which one of the following is an application of the backtracking algorithm?

A) Finding the shortest path

B) Finding the efficient quantity to shop

C) Ludo

D) Crossword

Answer: - D

7. Who coined the term 'backtracking'?

A) Lehmer

B) Donald

C) Ross

D) Ford

Answer: - A

8. The problem of finding a subset of positive integers whose sum is equal to a given positive integer is called as?

A) n- queen problem

B) Subset sum problem

C) Knapsack problem

D) Hamiltonian circuit problem

Answer: - B

9. The problem of placing n queens in a chessboard such that no two queens attack each other is called as?

A) n-queen problem

B) eight queens puzzle

C) four queens puzzle

D) 1-queen problem

Answer: - A

10. In how many directions do queens attack each other?

A) 1

B) 2

C) 3

D) 4

Answer: - C

11. Placing n-queens so that no two queens attack each other is called?

A) n-queen's problem

B) 8-queen's problem

C) Hamiltonian circuit problem

D) subset sum problem

Answer: - A

12. Where is the n-queens problem implemented?

A) carom

B) chess

C) ludo

D) cards

Answer: - B

13. Not more than 2 queens can occur in an n-queens problem.

A) true

B) false

Answer: - B

14. In n-queen problem, how many values of n does not provide an

optimal solution?

A) 1

B) 2

C) 3

D) 4

Answer: - B

15. Which of the following methods can be used to solve n-queen's problem?

A) greedy algorithm

B) divide and conquer

C) iterative improvement

D) backtracking

Answer: - D

16. Of the following given options, which one of the following is a correct option that provides an optimal solution for 4-queens problem?

A) (3,1,4,2)

B) (2,3,1,4)

C) (4,3,2,1)

D) (4,2,3,1).

Answer: - A

17. How many possible solutions exist for an 8-queen problem?

A) 100

B) 98

C) 92

D) 88

Answer: - C

18. How many possible solutions occur for a 10-queen problem?

A) 850

B) 742

C) 842

D) 724.

Answer: - D

19. The Knapsack problem is an example of _____

A) Greedy algorithm

B) 2D dynamic programming

C) 1D dynamic programming

D) Divide and conquer

Answer: - B

20. Which of the following methods can be used to solve the Knapsack problem?

A) Brute force algorithm

B) Recursion

C) Dynamic programming

D) Brute force, Recursion and Dynamic Programming

Answer: - D

21. You are given a knapsack that can carry a maximum weight of 60. There are 4 items with weights {20, 30, 40, 70} and values {70, 80, 90, 200}. What is the maximum value of the items you can carry using the knapsack?

A) 160

B) 200

C) 170

D) 90

Answer: - A

22. Which of the following problems is equivalent to the 0-1 Knapsack problem? A) You are given a bag that can carry a maximum weight of W. You are given N items which have a weight of {w1, w2, w3,...., wn} and a value of {v1, v2, v3,...., vn}. You can break the items into smaller pieces. Choose the items in such a way that you get the
maximum value

B) You are studying for an exam and you have to study N questions. The questions take {t1, t2, t3,...., tn} time(in hours) and carry {m1, m2, m3,...., mn} marks. You can study for a maximum of T hours. You can either study a question or leave it. Choose the questions in such a way that your score is maximized

C) You are given infinite coins of denominations {v1, v2, v3,....., vn} and a

sum S. You have to find the minimum number of coins required to get the sum S

D) You are given a suitcase that can carry a maximum weight of 15kg. You are given 4 items which have a weight of {10, 20, 15,40} and a value of {1, 2, 3,4}. You can break the items into smaller pieces. Choose the items in such a way that you get the maximum value

Answer: - B

23. What is the time complexity of the brute force algorithm used to solve the Knapsack problem?

A) O(n)

B) O(n!)

C) $O(2^n)$

D) $O(n^3)$

Answer: - C

24. Which of the following is/are property/properties of a dynamic programming problem?

A) Optimal substructure

B) Overlapping subproblems

C) Greedy approach

D) Both optimal substructure and overlapping subproblems

Answer: - D

25. If an optimal solution can be created for a problem by constructing optimal solutions for its subproblems, the problem possesses _____ property.

A) Overlapping subproblems

B) Optimal substructure

C) Memoization

D) Greedy

Answer: - B

26. If a problem can be broken into subproblems which are reused several times, the problem possesses _____ property.

A) Overlapping subproblems

B) Optimal substructure

C) Memoization

D) Greedy

Answer: - A

27. If a problem can be solved by combining optimal solutions to non-overlapping problems, the strategy is called _____

A) Dynamic programming

B) Greedy

C) Divide and conquer

D) Recursion

Answer: - C

28. In dynamic programming, the technique of storing the previously calculated values is called _____

A) Saving value property

B) Storing value property

C) Memoization

D) Mapping

Answer: - C

29. When a top-down approach of dynamic programming is applied to a problem, it usually _____

A) Decreases both, the time complexity and the space complexity

B) Decreases the time complexity and increases the space complexity

C) Increases the time complexity and decreases the space complexity

D) Increases both, the time complexity and the space complexity

Answer: - B

30. Which of the following problems is NOT solved using dynamic programming?

A) 0/1 knapsack problem

B) Matrix chain multiplication problem

C) Edit distance problem

D) Fractional knapsack problem

Answer: - D

31. Which of the following problems should be solved using dynamic programming?

A) Mergesort

B) Binary search

C) Longest common subsequence

D) Quicksort

Answer: - C

32. Time Complexity of Breadth First Search is? (V - number of vertices, E - number of edges)

A) O(V+E)

B) O(V)

C) O(E)

D) O(VE)

Answer: - A

33. The spanning tree of connected graph with 10 vertices contains .....

A) 9 edges

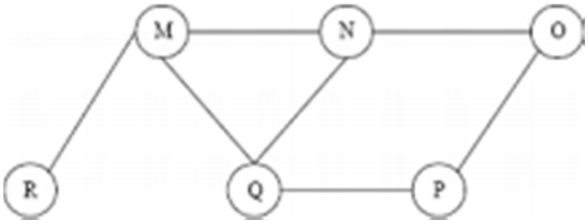
B) 11 edges

C) 10 edges

D) 8 edges

Answer: - A

34. The Breadth First Search algorithm has been implemented using the queue data structure. One possible order of visiting the nodes of the following graph is



A) MNOPQR

B) NQMPOR

C) QMNPRO

D) QMNPOR

Answer: - C

35. What is the maximum height of queue (To keep track of un-explored nodes) required to process a connected Graph G1 which contains 'N' node using BFS algorithm?

A) (N/2)-1

B) (N/2)/2

C) N-1

D) N

Answer: - C

PART B

1. What is meant by knapsack problem?
2. Define fractional knapsack problem.
3. Write the running time of 0/1 knapsack problem.
4. Write recurrence relation for 0/1 knapsack problem
5. What is meant by travelling salesperson problem?
6. What is the running time of dynamic programming TSP?
7. State if backtracking always produces optimal solution.
8. Define backtracking.
9. What are the two types of constraints used in backtracking?
10. What is meant by optimization problem?
11. Define Hamiltonian circuit problem.
12. What is Hamiltonian cycle in an undirected graph?
13. Define 8queens problem. 8. List out the application of backtracking.
14. Define promising node and non-promising node.
15. Give the explicit and implicit constraint for 8-queen problem.
16. How can we represent the solution for 8-queen problem?
17. Give the categories of the problem in backtracking.
18. Differentiate backtracking and over exhaustive search.
19. Find optimal solution for the knapsack instance n =3,w=[20,15,15],P =[40,25,25]and C =30
20. What is travelling salesperson problem?
21. What is the formula used to find upper bound for knapsack problem? 22. Differentiate between back tracking and branch and bound.
23. List out the application of branch and bound technique.
24. Analyze the time complexity for Warshall's and Floyd's algorithm. 25. Test the 0/1 knapsack problem.

26. Summarize Warshall's algorithm

27. Compare feasible and optimal solution.

28. Differentiate between DFS and BFS.

29. What is efficiency of DFS based algorithm for topological sorting

30. What are the different applications of DFS and BFS?

PART C

1. Describe the travelling salesman problem and discuss how to solve it using dynamic programming?

2. Find the optimal solution for the given knapsack problem

Item	1	2	3	4
weight	2	1	3	2
Value	$12	$10	$20	$15

.

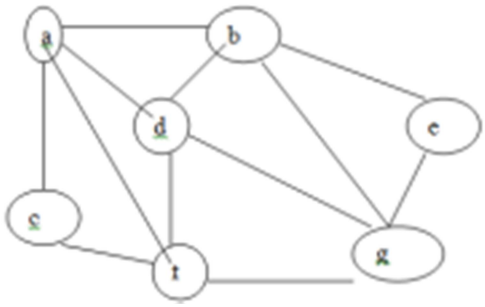
3. Apply backtracking technique to solve the following instance of the subset sum problem S = [1,3,4,5} and d=11 16

4. Explain subset-sum problem and discuss the possible solution strategies using backtracking.

5. Explain N-queens problem with an algorithm.

6. Explain why backtracking is defined as a default procedure of last resort for solving problems.

7. Explain the subset-sum problem in detail by justifying it using backtracking algorithm. 8. Apply backtracking to the problem of finding a Hamiltonian circuit for the following graph.



9. What is backtracking?Explain in detail.

10. Solve the following instance of the knapsack problem by the branch and bound algorithm.

Item	Weight	Value
1	4	$40
2	7	#42
3	5	$25
4	3	$12
The Knapsack's capacity W=10		

11. Discuss the solution for knapsack problem using branch and bound technique.

12. What is branch and bound technique?Explain how knapsack problem could be solved using branch and bound technique.Solve the following instance of the knapsack problem by branch and bound algorithm for W=16

Item	Weight	Value in Rs.
1	10	100
2	7	63
3	8	56
4	4	12

13. What is branch and bound? Explain in detail.

14. Consider the below matrix for assignment problem involving persons and jobs.Explain in detail how branch and bound technique is useful in solving assignment problems.

	Job1	Job2	Job3	Job4
A	9	2	7	8
B	6	4	3	7
C	5	8	1	8
D	7	6	9	4

15. Discuss Warshall's algorithm with suitable example.

UNIT V

Introduction to randomization and approximation algorithm - Randomized hiring problem - Randomized quick sort, Complexity analysis - String matching algorithm, Examples - Rabin Karp algorithm for string matching, Example discussion - Approximation algorithm, Vertex covering - Introduction Complexity classes, P type problems - Introduction to NP type problems, Hamiltonian cycle problem - NP complete problem introduction, Satisfiability problem - NP hard problems - Examples

PART A

1. What is a Rabin and Karp Algorithm?

(A) String Matching Algorithm
(B) Shortest Path Algorithm
(C) Minimum spanning tree Algorithm
(D) Approximation Algorithm
Answer: - A

2. What is the pre-processing time of Rabin and Karp Algorithm?

A) Theta(m^2)
B) Theta(mlogn)
C) Theta(m)
D) Big-Oh(n)
Answer: - C

3. Rabin Karp Algorithm makes use of elementary number theoretic notions.

A) True
B) FALSE
Answer: - A

4. Given a pattern of length- 5 window, find the spurious hit in the given text string.

Pattern: 3 1 4 1 5
Modulus: 13
Index: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
Text: 2 3 5 9 0 2 3 1 4 1 5 2 6 7 3 9 9 2 1 3 9
A**)** 6-10

B) 12-16
C) 3-7
D) 13-17
Answer: - D

5. What is the basic principle in Rabin Karp algorithm?
A) Hashing
B) Sorting
C) Augmenting
D) Dynamic Programming
Answer: - A

6. The worst-case efficiency of solving a problem in polynomial time is?
A)O(p(n))
B)O(p(nlogn))
C) O(p(n^2))
D) O(p(m log n))
Answer: - A

7. Problems that can be solved in polynomial time are known as?
A) Intractable
B) Tractable
C) Decision
D) Complete
Answer: - B

8. _____ is the class of decision problems that can be solved by non-deterministic polynomial algorithms.
A) NP
B) P
C) Hard
D) Complete
Answer: - A

9. The Euler's circuit problem can be solved in?
A) O(N)
B) O(N log N)
C)O(logN)
D) O(N^2)
Answer: - D

10. To which of the following class does a CNF-satisfiability problem belong?
A) NP class
B) P class
C) NP complete
D) NP hard

Answer: - C

11. Quick sort uses which of the following algorithm to implement sorting?

A) backtracking
B) greedy algorithm
C) divide and conquer
D) dynamic programming

Answer: - C

12. What is the worst case time complexity of randomized quicksort?

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

Answer: - C

13. What is the purpose of using randomized quick sort over standard quick sort?

A) so as to avoid worst case time complexity
B) so as to avoid worst case space complexity
C) to improve accuracy of output
D) to improve average case time complexity

Answer: - A

14. Which of the following is incorrect about randomized quicksort?

A) it has the same time complexity as standard quick sort
B) it has the same space complexity as standard quick sort
C) it is an in-place sorting algorithm
D) it cannot have a time complexity of O(n2) in any case.

Answer: - D

15. Which of the following is the fastest algorithm in string matching field?

A) Boyer-Moore's algorithm
B) String matching algorithm
C) Quick search algorithm
D) Linear search algorithm

Answer: - C

16. What is vertex coloring of a graph?

A) A condition where any two vertices having a common edge should not have same color
B) A condition where any two vertices having a common edge should always have same color

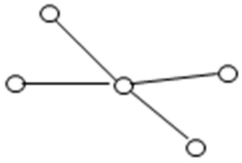
C) A condition where all vertices should have a different color
D) A condition where all vertices should have same color
Answer: - A

17. How many edges will a tree consisting of N nodes have?
A) Log(N)
B) N
C) N-1
D) N+1
Answer: - C

18. Minimum number of unique colors required for vertex coloring of a graph is called?
A) vertex matching
B) chromatic index
C) chromatic number
D) color number.
Answer: - C

19. How many unique colors will be required for proper vertex coloring of an empty graph having n vertices?
A) 0
B) 1
C) n
D) n!
Answer: - C

20. What will be the chromatic number of the following graph?



A) 1
B) 2
C) 3
D) 4
Answer: - B

21. Assuming P != NP, which of the following is true ?
A) NP-complete = NP
B) NP-complete \cap P = \Phi
C) NP-hard = NP
D) P = NP-complete
Answer: - B

22. Let X be a problem that belongs to the class NP. Then which one of the

following is TRUE?

A) There is no polynomial time algorithm for X.
B) If X can be solved deterministically in polynomial time, then P = NP.
C) If X is NP-hard, then it is NP-complete.
D) X may be undecidable.
NP Complete
Answer: - C

23. Which of the following statements are TRUE?

1. The problem of determining whether there exists a cycle in an undirected graph is in P.
2. The problem of determining whether there exists a cycle in an undirected graph is in NP.
3. If a problem A is NP-Complete, there exists a non-deterministic polynomial time algorithm to solve A.

A) 1, 2 and 3
B) 1 and 2 only
C) 2 and 3 only
D) 1 and 3 only
Answer: - A

24. Consider the following two problems on undirected graphs
α : Given G(V, E), does G have an independent set of size | V | - 4?
β : Given G(V, E), does G have an independent set of size 5?
Which one of the following is TRUE?

A) α is in P and β is NP-complete
B) α is NP-complete and β is in P
C) Both α and β are NP-complete
D) Both α and β are in P
Answer: - C

25. Which of the following algorithm can be used to solve the Hamiltonian path problem efficiently?

A) branch and bound
B) iterative improvement
C) divide and conquer
D) greedy algorithm
Answer: - A

26. Hamiltonian path problem is _____

A) NP problem

B) N class problem
C) P class problem
D) NP complete problem
Answer: - D

27. There is no existing relationship between a Hamiltonian path problem and Hamiltonian circuit problem.

A) true
B) false
Answer: - B

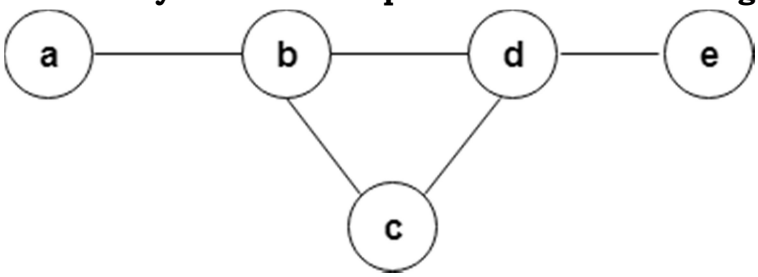
28. Which of the following problems is similar to that of a Hamiltonian path problem?

A) knapsack problem
B) closest pair problem
C) travelling salesman problem
D) assignment problem
Answer: - C

29. In what time can the Hamiltonian path problem can be solved using dynamic programming?

A) O(N)
B) O(N log N)
C) O(N2)
D) O(N2 2N)
Answer: - D

30. How many Hamiltonian paths does the following graph have?



A) 1
B) 2
C) 3
D) 4
Answer: - A

31. A node is said to be _____ if it has a possibility of reaching a complete solution.

A) Non-promising
B) Promising
C) Succeeding
D) Preceding
Answer: - B

32. Minimum number of unique colors required for vertex coloring of a graph is called?
A) vertex matching
B) chromatic index
C) chromatic number
D) color number
Answer: C

PART B

1. Define NP hard and NP completeness.
2. Compare NP hard and NP completeness.
3. Write Short notes on “the class P and NP problem”.
4. How NP Hard problems are different from NP Complete?
5. Whether class P solves a problem in polynomial time? Justify.
6. An NP hard problem can be solved in deterministic polynomial time, how?
7. Give examples for NP Complete problems
8. State the property of NP complete problem.
9. Define adversary method.
10. Define lower bound.
11. What type of output yields trivial lower bound?
12. What is information theoretic lower bound?
13. Define complexity theory.
14. What is halting problem?
15. What is CNFs satisfiablity problem?
16. Define Matching.
17. Define a bipartite graph.
18. How will you check the stability?
19. What is stable marriage problem?
20. Define the term stable pair
21. What do you mean by perfect match in bipartite graph?
22. Write Rabin Karp string matching algorithm
23. Describe Hamiltonian cycle problem

PART C

1. Describe in detail about P and NP Problems
2. Write short notes on NP Complete Problem
3. Write short notes on the following using approximation Algorithm

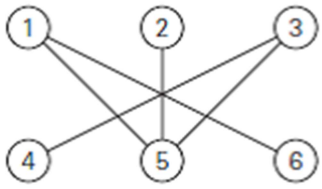
i) Nearest –neighbor algorithm with example
ii) Multi fragment heuristic algorithm with example

4. Describe in detail about Twice around the tree algorithm with example
5. Explain local search heuristic with example
6. Explain Approximation Algorithms for the Travelling Salesman Problem
7. Explain the Assignment problem in Branch and bound with Example.
8. Suggest an approximation algorithm for TSP. Assume that the cost function satisfies the triangle inequality.
9. Using an example prove that, satisfiability of Boolean formula in 3-Conjunctive Normal Form is NP – complete.
10. State the relationships among the complexity class algorithms with the help of neat diagrams
11. Explain the algorithm for stable marriage problem and prove the theorem with Example
12. Consider an instance of the stable marriage problem given by the ranking matrix

	A	**B**	**C**
α	1,3	2,2	3,1
β	3,1	1,3	2,2
γ	2,2	3,1	1,3

For each of its marriage matching's, indicate whether it is stable or not

13. Apply the maximum matching algorithm to the following bipartite graphs



14. Write the algorithm for maximum matching in Bipartite Graphs and prove the theorem With example
15. Explain the algorithm:
    i. Blocking pair
    ii. Stable marriage problem
    iii. Man optimal
    iv. Women optimal
16. Explain briefly on minimum weight perfect matching algorithm.
17. Explain briefly on reducing bipartite graph to net flow
18. Explain local search heuristic with example
19. Consider the following minimization problem:
    DEGREE BOUNDED SPANNING TREE:
    Instance: Graph G = (V,E)

Solution:: A spanning tree T of G
Value: Maximum degree of T
Goal: Find a solution with minimum value.

20. Consider the following scheduling problem. You are given n jobs where job i is specied by an earliest start time si and a processing time pi. In homework 1, we considered a preemptive version of this problem and gave a greedy algorithm to give an optimal preemptive schedule. In this problem we consider the non-preemptive version of this scheduling problem. Here a job CANNOT be suspended but rather must be performed in a contiguous time interval. Consider the following heuristic for the non-preemptive problem: schedule the jobs in the order in which they complete in an optimal preemptive schedule starting each job as soon as the one before it completes. You are to prove that this algorithm is a 2-approximation algorithm.
21. Using Rabin karp string matching algorithm match the given pattern P with given string S.
    P = 745
    S = 745727457
22. Using KMP string matching algorithm, find the occurrence of the given patter P in the given text T.
    T ß ABABACAB
    P ß ABAB