Chettinad College of Engineering & Technology
Approved by AICTE-New Delhi and Affiliated to Anna University-Chennai

Academic Year 2024 - 2025

Question Bank

Year/Semester: II/ III **Date: / /2024**	**Department: AI & DS** **Subject Code/Title :AD3351 –Design and Analysis of Algorithms** **Faculty Name:Mrs.C.Sangeetha**	**Unit:** I/II/III/IV/V **Section: Part A/B/C**

UNIT I
Part A

1. **State the transpose symmetry property of O and Ω. [Nov/Dec 2019]**
   f(n) = O(g(n)) if and only if g(n) = Ω(f(n))

2. **Define recursion. [Nov/Dec 2019]**
   The process in which a function calls itself directly or indirectly is called recursion and the corresponding function is called as recursive function. Using recursive algorithm, certain problems can be solved quite easily

3. **How do you measure the efficiency of an Algorithm? [Apr/May 2019]**
   - Size of the Input
   - Running Time

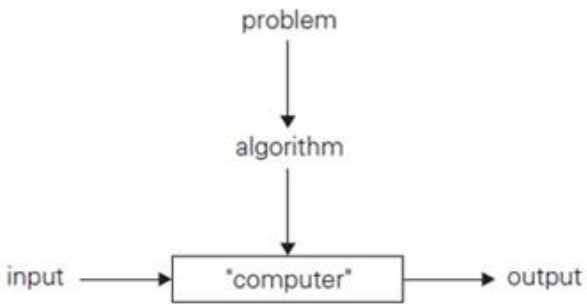
4. **Prove that f(n)= O(g(n)) and g(n)=O(f(g(n)) then f(n)=Θ(g(n)) [Apr/May 2019]**
   Prove by contradiction. Assume f(n) = O(g(n)), by the definition, there exist constants c, n0 > 0 such that $0 \le f(n) \le cg(n)$ or $0 \le n \le cn1+\sin n$ for all $n \ge n0$. It implies (0.6) $0 \le 1 \le cn\sin n$ for all $n \ge n0$. Can it be true? To show that the answer is No, it suffices to show: For any n0 > 0, we can always pick an $n \ge n0$ such that $cn\sin n < 1$.

5. **What is basic operation ?[Apr/May 2018]**

   The operation that contributes most towards the running time of the algorithm The running time of an algorithm is the function defined by the number of steps (or amount of memory) required to solve input instances of size n.

6. **What is an Algorithm? [Apr/May 2017]**

An algorithm is a sequence of unambiguous instructions for solving a problem. i.e., for obtaining a required output for any legitimate input in a finite amount of time



7. How to measure algorithm's running time? [Nov/Dec 2017]

Time efficiency indicates how fast the algorithm runs. An algorithm's time efficiency is measured as a function of its input size by counting the number of times its basic operation (running time) is executed. Basic operation is the most time consuming operation in the algorithm's innermostloop.

8.Compare the order of growth n(n-1)/2 and n^2. [May/June2016]

n	n(n-1)/2	n^2
Polynomial	**Quadratic**	**Quadratic**
1	0	1
2	1	4
4	6	16
8	28	64
10	45	10^2
10^2	4950	10^4
Complexity	Low	High
Growth	Low	high

n(n-1)/2 is lesser than the half of n^2

9. Define recurrence relation. [Nov/Dec2016]

A recurrence relation is an equation that defines a sequence based on a rule that gives the next term as a function of the previous term(s). The simplest form of a recurrence relation is the case where the next term depends only on the immediately previous term.

10. Write down the properties of asymptotic notations. [April/May2015]

Asymptotic notation is a notation, which is used to take meaningful statement about the efficiency of a program. To compare and rank such orders of growth, computer scientists use three notations, they are:

- O - Big ohnotation
- Ω - Big omeganotation
- Θ - Big thetanotation

11. Give the Euclid's Algorithm for Computing gcd(m,n) [Apr/May `16,`18]

Algorithm *Euclid_gcd(m,n)*

//Computes gcd*(m, n)* by Euclid's algorithm

//Input: Two nonnegative, not-both-zero integers *m* and *n*

//Output: Greatest common divisor of *m* and *n*

while $n \neq 0$ **do**

$r \leftarrow m \bmod n$

m←n

n←r

return *m*

Example: gcd(60, 24) = gcd(24, 12) = gcd(12, 0) = 12.

12. Write an algorithm to find the number of binary digits in the binary representation ofapositive decimal integer. [Apr May 2015]

Algorithm *Binary(n)*

//Input: A positive decimal integer *n*

//Output: The number of binary digits in *n*'s binary representation

count ←1

while *n* >1 **do**

count ←*count* + 1

n←[*n*/2]

return *count*

13. What are the processes in algorithmic problem solving? [Nov/Dec2009]

- Understanding the problem
- Decision making on - Capabilities of computational devices, Choice of exact orapproximate problem solving, Data structures
- Algorithmic design techniques
- Design an algorithm
- Prove correctness
- Analyze the algorithms

14. Define Big Theta Notations [Nov/Dec2014]

A function t(n) is said to be in Θ(g(n)) , denoted t(n) Є Θ (g(n)) , if t(n) is bounded both above and below by some positive constant multiple of g(n) for all large n, i.e., if there exist some positive constants c1 and c2 and some nonnegative integer n0 such that c1

g(n) ≤t(n) ≤ c2g(n) for all n ≥ n0

15. Define Big Omega Notations. [May/June2013]

A function t(n) is said to be in Ω(g(n)) , denoted t(n) Є Ω((g(n)) , if t(n) is boundedbelow by some positive constant multiple of g(n) for all large n, i.e., if there exist some positive constant c and some nonnegative integer n0 such that

t(n) ≥cg(n) for all for all n ≥ n0

16.What is Big 'Oh' notation? [May/June2012]

A function t(n) is said to be in O(g(n)), denoted t(n) O(g(n)) , if t(n) is bounded above by some constant multiple of g(n) for all large n, i.e., if there exist some positive constant c and some nonnegative integers no such that

t(n) ≤ cg(n) for all n≥ n0

17. Give the two major phases of performance evaluation.

Performance evaluation can be loosely divided into two major phases:

a. a prior estimates (performanceanalysis)

b. a Posterior testing (performancemeasurement)

18. **List the factors which affects the running time of the algorithm.**

A. Computer

B. Compiler

C. Input to the algorithm

i. The content of the input affects the running time

ii. Typically, the input size is the main consideration.

19. Give an non-recursive algorithm to find out the largest element in a list of n numbers.

ALGORITHM MaxElement(A[0..n-1])
//Determines the value of the largest element in a given array
Input:An array A[0..n-1] of real numbers
//Output: The value of the largest element in A
maxvalï*f*Ÿa[0] for I ï*f*Ÿ 1 to n-1 do
if A[I] >maxval return maxvalï*f*Ÿ A[I] return maxval

20. Write a recursive algorithm for computing the nth fibonacci number.

ALGORITHM F(n)
// Computes the nth Fibonacci number recursively by using the definition
// Input A non-negative integer n
// Output The nth Fibonacci number
if n 1 return n
else return F(n-1)+F(n-2)

21. What is algorithm visualization?

Algorithm visualization can be defines as the use of images to convey some useful information about algorithms. Two principal variations are Static algorithm visualization Dynamic Algorithm visualization(also called algorithm animation)

22.What is the order of growth?

The Order of growth is the scheme for analyzing an algorithm's efficiency for different input sizes which ignores the multiplicative constant used in calculating the algorithm's running time. Measuring the performance of an algorithm in relation with the input size n is called the order of growth.

Order of Growth	Name
$O(1)$	constant
$O(\log_b n)$	logarithmic (for any b)
$O(n)$	linear
$O(n \log_b n)$	"en log en"
$O(n^2)$	quadratic
$O(n^3)$	cubic
$O(c^n)$	exponential (for any c)

Part B & C- (16 marks & 8 marks)

1. Explain about algorithm with suitable example
2. Write an Algorithm using recursion that determines the GCD of two numbers. Determine the time and space complexity.
3. Discuss about Fundamentals of Algorithmic Problem Solving.
4. Discuss important problem types that you face during Algorithm Analysis.
5. Discuss Fundamentals of the analysis of algorithm efficiency elaborately
6. Elaborate asymptotic analysis of an algorithm with an example.
7. Discuss about Mathematical Analysis of non-recursive Algorithms.(Searching, matrix multiplication, maximum value in a given array, Whether all elements are distinct or not,)
8. Explain in detail about Mathematical Analysis of Recursive Algorithms.(factorial, fibonocci, tower of Hanoi problem)
9. Explain in detail about linear search with examples
10. Solve the recurrence relation T(n)= 2T(n-2)+6T(n-2),T(n)=(n/3)+C
11. Explain Big oh, Big omega and Big Theta Notation with examples.

UNIT II
Part A

1. State the Convex Hull Problem.

The convex hull of a set of points is defined as the smallest convex polygon, that encloses all of the points in the set. Convex means that the polygon has no corner that is bent inwards.

2. Write the Brute force algorithm to string matching.[Apr/May 2019]

ALGORITHM *BruteForceStringMatch*$(T[0..n-1], P[0..m-1])$

```
//Implements brute-force string matching
//Input: An array T[0..n − 1] of n characters representing a text and
//       an array P[0..m − 1] of m characters representing a pattern
//Output: The index of the first character in the text that starts a
//        matching substring or −1 if the search is unsuccessful
for i ← 0 to n − m do
    j ← 0
    while j < m and P[j] = T[i + j] do
        j ← j + 1
    if j = m return i
return −1
```

3. **What is the time and space complexity of Merge Sort? [Apr/May 2019]**

Time Complexity	Space Complexity
Best Case: Θ(n log n)	Best Case: Θ(n log n)
Average Case: Θ(n log n)	Average Case: n log n
Worst Case: Θ(n log n)	Worst Case: n log n

4. **What is exhaustive search?[Apr/May 2018]**

An Exhaustive Search, also known as generate and test, is a very general problem solving technique that consist of systematically enumeration all possible candidates for the solution and checking whether each candidate satisfies the problem's statement.

5. **State Master's Theorem. [Apr/May 2018]**

Let T(n) be a monotonically increasing function tsatisfies
$T(n) = aT(n/b) + f(n)$ $T(1) = c$
Where $a \geq 1, b \geq 2, c > 0$. *If f(n) ∈Θ($n^d$) where d ≥ 0, then*

$$T(n) \in \begin{cases} \Theta(n^d) & \text{if } a < b^d, \\ \Theta(n^d \log n) & \text{if } a = b^d, \\ \Theta(n^{\log_b a}) & \text{if } a > b^d. \end{cases}$$

6. **What is Closest Pair Problem? [May/June 2016, Apr/May 2017]**

The closest-pair problem finds the two closest points in a set of n points. It is the simplest of a variety of problems in computational geometry that deals with proximity of points in the plane or higher- dimensional spaces. The distance between two Cartesian coordinates is calculated by Euclidean distance formula

$$d(p_i, p_j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$$

---

7. Give the General Plan for Divide and Conquer Algorithms

A divide and conquer algorithm works by recursively breaking down a problem into two or more sub-problems of the same (or related) type (divide), until these become simple enough to be solved directly (conquer). Divide-and-conquer algorithms work according to the following general plan:

- A problem is divided into several subproblems of the same type, ideally of about equalsize.
- The subproblems are solved (typically recursively, though sometimes a different algorithm is employed, especially when subproblems become smallenough).
- If necessary, the solutions to the subproblems are combined to get a solution to theoriginal problem.

Example: Merge sort, Quick sort, Binary search, Multiplication of Large Integers and Strassen's Matrix Multiplication.

8. Write the advantages of insertion sort. [Nov/Dec 2017]

- Simple implementations
- Efficient for Small Data Sets
- Stable
- More efficient
- Online

9. Derive the Complexity of Binary Search [Apr/May 2015]

In conclusion we are now able completely describe the computing time of binary search by giving formulas that describe the best, average and worst cases.

Successful searches	Unsuccessful searches
Best case - $\Theta(1)$ Average case - $\Theta(\log_2 n)$ Worst case - $\Theta(\log_2 n)$	Best case, Average case, Worst case - $\Theta(\log_2 n)$

---

10. Write about traveling sales person problem.

Let g = (V, E) be a directed. The tour of G is a directed simple cycle that includes every vertex in V. The cost of a tour is the sum of the cost of the edges on the tour. The traveling salesperson problem to find a tour of minimum cost.

11. What is binary search?

Binary search is a remarkably efficient algorithm for searching in a sorted array. It works by comparing a search key K with the arrays middle element A[m]. if they match the algorithm stops; otherwise the same operation is repeated recursively for the first half of the array if K < A[m] and the second half if K > A[m].

K >A[0].....A[m-1]A[m]A[m+1] A[n-1]

search here if KA[m]

12. What is Knapsack problem?

A bag or sack is given capacity and n objects are given. Each object has weight wi and profit pi. Fraction of object is considered as xi (i.e) 0<=xi<=1 .If fraction is 1 then entire object is put into sack. When we place this fraction into the sack, we get wi xi and pi xi.

13. Write an algorithm for brute force closest-pair problem.

Algorithm BruteForceClosestPair(P)
//Finds distance between two closest points in the plane by brute force
//Input: A list P of n (n ≥ 2) points p1(x1, y1), . . . ,pn(xn, yn)
//Output: The distance between the closest pair of points
d←∞
for i←1 to n − 1 do
for j ←i+ 1 to ndo
d ←min(d, sqrt((xi − xj)2 + (yi− yj)2)) //sqrt is square root
return d

14. Define extreme points.

An extreme point of a convex set is a point in the set that does not lie on any open line segment between any other two points of the same set. For a convex hull, every extreme point must be part of the given set, because otherwise it cannot be formed as a convex combination of given points.

15. What is an exhaustive search?

Exhaustive Search is a brute-force algorithm that systematically enumerates all possible solutions to a problem and checks each one to see if it is a valid solution. This algorithm is typically used for problems that have a small and well-defined search space, where it is feasible to check all possible solutions.

Part B & C- (16 marks & 8 marks)

1. Discuss Quick Sort Algorithm and Explain it with example. Derive Worst case and Average Case Complexity.
2. Explain Merge Sort with suitable example.
3. Explain in detail about Travelling Salesman Problem using exhaustive search.
4. Explain in detail about Knapsack Problem and Assignment Problem.
5. Write algorithm to find closest pair of points using divide and conquer and explain it withexample. Derive the worst case and average case time complexity.
6. What is Convex hull problem? Explain the brute force approach to solve convex-hull with an example. Derive time complexity.
7. Explain about Strassen's Matrix Multiplication with an example problem.
8. Discuss about the Decrease and Conquer method and perform Topological Sorting by performing DFS.
9. Elaborate the Transform and Conquer methodology and discuss about Presorting, Heaps and Heap Sort.

UNIT III

Part A

1. State the Principle of Optimality.

The principle of optimality is the basic principle of dynamic programming. It states that an optimal sequence of decisions or choices, each subsequence must also be optimal.

2. What is the Constraint for binary search tree for insertion?

When inserting or searching for an element in a binary search tree, the key of each visited node has to be compared with the key of the element to be inserted or found. The shape of the binary search tree depends entirely on the order of insertions and deletions, and can become degenerate.

3. Define multistage graph. Give Example.

A Multistage graph is a directed graph in which the nodes can be divided into a set of stages such that all edges are from a stage to next stage only (In other words there is no edge between vertices of same stage and from a vertex of current stage to previous stage).

4. How Dynamic Programming is used to solve Knapsack Problem?

An example of dynamic programming is Knapsack problem. The solution to the Knapsack problem can be viewed as a result of sequence of decisions. We have to decide the value of xi for 1<i<n. First we make a decision on x1 and then on x2 and so on. An optimal sequence of decisions maximizes the object function Σpixi.

5. Define transitive closure of directive graph.

The transitive closure of a directed graph with 'n' vertices is defined as the n-by-n Boolean matrix T={tij}, in which the elements in the ith row (1 i n) and the jth column (1 j n) is 1 if there exists a non trivial directed path from the ith vertex to the jth vertex otherwise, tij is 0 .

6. Define the Minimum Spanning tree problem

A minimum spanning tree (MST) or minimum weight spanning tree is a subset of the edges of a connected, edge-weighted (un)directed graph that connects all the vertices together, without any cycles and with the minimum possible total edge weight.

7. What does Floyd's Algorithm do?

Floyd's algorithm is an application, which is used to find the entire pairs shortest paths problem. Floyd's algorithm is applicable to both directed and undirected weighted graph, but they do not contain a cycle of a negative length.

8. State the Assignment Problem

There are n people who need to be assigned to execute n jobs as one person per job. Each person is assigned to exactly one job and each job is assigned to exactly one person.

9.Define the Single Source Shortest Path Problem. [May/June 2016]

Single source shortest path problem can be used to find the shortest path from single source to all other vertices.

Example:Dijikstras algorithm

10. List out the memory function under dynamic programming.

- Top-Down Approach
- Bottom –Up Approach

11. What is Huffman trees?

A Huffman tree is binary tree that minimizes the weighted path length from the root to the leaves containing a set of predefined weights. The most important application of Huffman trees are Huffman code.

12. List the advantage of Huffman's encoding?

- Huffman's encoding is one of the most important file compression methods.
- It is simple
- It is versatility
- It provides optimal and minimum length encoding

13. What do you mean by Huffman code?

A Huffman code is a optimal prefix tree variable length encoding scheme that assigns bit strings to characters based on their frequencies in a given text.

14. What is greedy method?

The greedy method is the most straight forward design, which is applied for change making problem.

The greedy technique suggests constructing a solution to an optimization problem through a sequence of steps, each expanding a partially constructed solution obtained so far, until a complete solution to the problem is reached. On each step, the choice made must be feasible, locally optimal and irrevocable.

15. **What do you mean by row major and column major?**

In a given matrix, the maximum elements in a particular row is called row major.

In a given matrix, the maximum elements in a particular column is called column major.

16. Compare Greedy method and Dynamic Programming

Greedy method	Dynamic programming
1.Only one sequence of decision is generated.	1.Many number of decisions are generated.
2.It does not guarantee to give an optimal solution always.	2.It definitely gives an optimal solutionalways.

17. 17. Show the general procedure of dynamic programming. [APR/MAY 2017]

The development of dynamic programming algorithm can be broken into a sequence of 4 steps.

- Characterize the structure of an optimalsolution.
- Recursively define the value of the optimalsolution.
- Compute the value of an optimal solution in the bottom-up fashion.
- Construct an optimal solution from the computed information

18. What is Multistage graph?

A Multistage graph is a directed graph in which the nodes can be divided into a set of stages such that all edges are from a stage to next stage only (In other words there is no edge between vertices of same stage and from a vertex of current stage to previous stage) i.e it is used to find a minimum cost path from source 's' to sink't'

19. List the features of dynamic programming?

Optimal solutions to sub problems are retained so as to avoid recomputing their values. Decision sequences containing subsequences that are sub optimal are not considered. It definitely gives the optimal solution always.

20.What do you mean by dynamic programming? (May 17)

Dynamic programming is a technique for solving problems with overlapping sub problems. Rather than solving overlapping sub problems again and again, dynamic programming suggests solving each of the smaller sub problems only once and recording the results in a table from which a solution to the original problem can then be obtained.

21.What does dynamic programming have in common with divide and conquer?

Both the techniques are based on dividing a problems instance into smaller instances of the same problem.

22. What is the difference between dynamic programming with divide and conquer method?

Divide and conquer divides an instance into smaller instances with no intersections whereas dynamic

programming deals with problems in which smaller instances overlap. Consequently divide and conquer algorithm do not explicitly store solutions to smaller instances and dynamic programming algorithms do.

23. **Define code word.**

Encode a text that comprises symbols from some n-symbol alphabet by assigning to each of the text's symbols some sequence of bits called the code word.

Part B & C- (16 marks & 8 marks)

1. Explain about Dynamic programming and Principle of optimality.
2. Write short notes on Coin changing problem.
3. Discuss about Warshall's algorithms with an example.
4. Define Optimal Binary Search Trees. Compute cost table and root table for (do,if,int,while) with probability (0.1,0.2,0.4,0.3)
5. Discuss about Multi stage graph.
6. Explain about Greedy Technique and perform Dijkstra's algorithm with source code.
7.Let A={ l/119, m/96, c/247, g/283, h/72, f/77,k/92, j/19} be the letters and its frequency of distribution in a text file. Compute a suitable Huffman coding to compress the data effectively.
8.Compute 0/1 Knapsack problem for the following:

Item	Weight	Value
1	2	3
2	3	4
3	4	5
4	5	6

The capacity of knapsack is W=5

9. Using dynamic programming, solve the following knapsack instance:
N=3,$[w_1,w_2,w_3]$=[1,2,2] and $[p_1,p_2,p_3]$=[18,16,6] and M=4
10. Solve all pair shortest path problem for the diagraph with the weight matrix given below using floyds algorithm.

	A	B	C	D
A	0	∞	∞	3
B	2	0	∞	∞
C	∞	7	0	1
D	6	∞	∞	0

UNIT IV

Part A

1.What is the purpose of iterative improvement technique?

The iterative-improvement technique involves finding a solution to an optimization problem by generating a sequence of feasible solutions with improving values of the problem's objective function. Each subsequent solution in such a sequence typically involves a small, localized change in the previous feasible solution. When no such change improves the value of the objective function, the algorithm returns the last feasible solution as optimal and stops.

2. List out the problems that can be solved by iterative improvement algorithms.

- **Linear programming (simplex method)**
- **Maximizing the flow in a network,**
- **Stable marriage problem**
- **Matching the maximum possible number of vertices in a graph**

3. Define simplex method.

The simple method is the classic method for solving the general linear programming problem. It works by generating a sequence of adjacent extreme points of the problem's feasible region with improving values of the objective function.

4. Define flow.
A (feasible) flow is an assignment of real numbers xij to edges (i, j) of a given network that satisfy flow- conservation constraints and the capacity constraints $0 \leq xij \leq uij$ for every edge (i, j) € E.

5.Define Ford-Fulkerson method.
The Ford-Fulkerson method is a classic template for solving the maximum flow problem by the iterative- improvement approach. The shortest augmenting-path method implements this idea by labeling network vertices in the breadth-first search manner.

6.Define forward edge.
An undirected graph in which any two consecutive vertices i, j are either connected by a directed edge from i to j with some positive unused capacity rij= uij− xij. Edges of this kind are called forward edges
because their tail is listed before their head in the vertex list 1→. . . i →j . . .→n defining the path.

7.Define backward edge.
A undirected graph in which any two consecutive vertices i, j are either connected by a directed edge from j to i with some positive flow xji. Edges of this kind are called backward edges because their tail is listed after their head in the path list
1→. . . i ←j . . .→n.

8.Define Max-Flow Min-Cut theorem.
Max-Flow Min-Cut theorem is defined as the value of a maximum flow in a network is equal to the
capacity of its minimum cut.

9.What you mean by matching in a graph.
A matching in a graph is a subset of its edges with the property that no two edges share a vertex.

10.Define maximum-matching problem.
The maximum-matching problem is the problem of finding a maximum matching (matching with the largest number of edges) in a given graph.

11.What is mean by bipartite graph? (or) Define 2-colorable in bipartite graph.
In a bipartite graph, all the vertices can be partitioned into two disjoint sets V and U, not necessarily of the same size, so that every edge connects a vertex in one of these sets to a vertex in the other set. In other words, a graph is bipartite if its vertices can be colored in two colors so that every edge has its vertices
colored in different colors; such graphs are also said to be 2-colorable.

12.What do you mean by maximum weight matching?
The problem of maximizing the sum of the weights on edges connecting matched pairs of vertices. This
problem is called maximum-weight matching.

13.When marriage matching problem is said to stable?
A marriage matching M is called stable if there is no blocking pair for it.

14.What is mean by stable marriage problem?
Consider a set Y = {m1, m2. . . mn} of n men and a set X = {w1, w2, . . . , wn} of n women. Each man has a
preference list ordering the women as potential marriage partners with no ties allowed. Similarly, each woman has a preference list of the men, also with no ties.

15.Define marriage matching.
A marriage matching M is a set of n (m, w) pairs whose members are selected from disjoint n-element sets Y and X in a one-one fashion, i.e., each man m from Y is paired with exactly one woman w from X and vice versa.

16.Write the algorithm for stable marriage problem.
Input: A set of n men and a set of n women along with rankings of the women by each man and rankings of the men by each woman with no ties allowed in the rankings
Output: A stable marriage matching
- Start with all the men and women being free.
- While there are free men, arbitrarily select one of them and do the following:

Proposal: The selected free man m proposes to w, the next woman on his preference list (who is the highest-ranked woman who has not rejected him before).

Response: If w is free, she accepts the proposal to be matched with m. If she is not free, she compares m with her current mate. If she prefers m to him, she accepts m's proposal, making her former mate free; otherwise, she simply rejects m's proposal, leaving m free. Return the set of n matched pairs.

17.Define blocking pair.

A pair (m, w), where m € Y, w € X, is said to be a blocking pair for a marriage matching M if man m and woman w are not matched in M but they prefer each other to their mates in M.

18.When marriage matching problem is said to unstable?

A marriage matching M is called stable if there is a blocking pair for it.

Prove that the stable marriage algorithm terminates after no more than n2 iterations with a stable marriage output.

The algorithm starts with n men having the total of n2 women on their ranking lists. On each iteration, one man makes a proposal to a woman. This reduces the total number of women to whom the men can still propose in the future because no man proposes to the same woman more than once. Hence, the algorithm must stop after no more than n2 iterations.

Prove that the stable marriage algorithm always yields a gender-optimal stable matching.

To prove that a man (woman)-optimal matching is unique for a given set of participant preferences. Therefore the algorithm's output does not depend on the order in which the free men (women) make their proposals.

19. What do you mean by 'Perfect Matching' in bipartite graphs?

- A perfect matching is a matching in which each node has exactly one edge incident on it. A Bipartite Graph G=(V,E) is a graph in which the vertex set V can be divided into two disjoint subsets X and Y such that every edge e € E has one end point in X and the other end point in Y.
- A matching M is a subset of edges such that each node in V appears in at most one edge in M.

20. Define flow 'cut'.

If all the edges of a cut where deleted from the network, there would be no directed path from source to sink.

21.What is Skew symmetry?

Skew symmetry constraint: The flow on an arc from u to v is equivalent to the negation of the flow on the arc from v to u, that is: $f(u, v) = -f(v, u)$. The sign of the flow indicates the flow's direction. Capacity constraint: An arc's flow cannot exceed its capacity, that is: $f(u, v) \leq c(u, v)$.

22.define residual network.

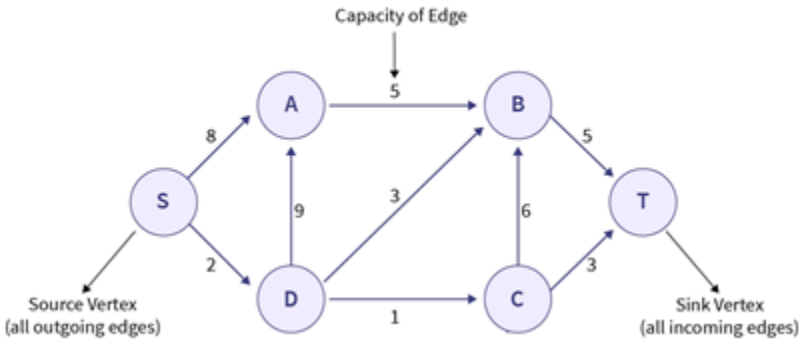
A residual network graph indicates how much more flow is allowed in each edge in the network graph. If there are no augmenting paths possible from to , then the flow is maximum. The result i.e. the maximum flow will be the total flow out of source node which is also equal to total flow in to the sink node.

23.**What is augmenting path?**

Augmenting path is a directed path from a source node s to a sink node t in the residual network. The residual capacity of an augmenting path is the minimum residual capacity of any arc in the path. Obviously, we can send additional flow from the source to the sink along an augmenting path.

Part B & C- (16 marks & 8 marks)

1. Explain about The Simplex Method by using iterative method with algorithm.
2. Discuss about The Maximum-Flow Problem with Ford Fulkerson algorithm.
3. Define Bipartite graph.Explain about Maximum Matching in Bipartite Graphs
4. Discuss about the Stable marriage Problem with examples.
5. Maximize p=2x+3y+z, Subject to x+y+z≤ 40, 2x+y-z ≥ 10, -y+ z ≥ 10 and x≥ 0,y≥0, z≥ 0
6. Solve the linear programing problem geometrically, Maximize 3x+ y, subject to : x+y=1,2x+y <=4,x>0,y>=0
7. A potter is making cups and plates. It takes her 6 minutes to make a cup and 3 minutes to make a plate. Each cup uses ¾ pounds of clay and each plate uses one pound of clay. She has 20 hours available for making the cups and plates and has 250 pounds of clay on hand. She makes a profit of $2 on each cup and $ 1.5 on each plate. How many cups and how many plates should she make in order to maximize her profit?
8. How do you compute a maximum flow for the following graph using ford Fulkerson method?



09. In a survey of color choice, 5 students are examined for their color choices. Following table shows the liking for particular colors shown by a student.

Student	**Color**
Ankita	Red,Blue
Prajakta	Green,Yellow
Supriya	Red,Green
Urmila	Blue,White
Varsha	White,Blue,Yellow

a)Draw a bipartite graph to model this situation
b)Using this matching use the maximum matching algorithm to find the complete matching.

UNIT V
Part A

1**. Define trivial lower bound of a class. (May 18)**
The simplest method of obtaining a lower-bound class is based on counting the number of items in the problem's input that must be processed and the number of output items that need to be produced. Since any algorithm must at least "read" all the items it needs to process and "write" all its outputs, such a count yields a trivial lower bound.

2**. Define tournament tree.**
A tournament tree is a complete binary tree reflecting results of a "knockout tournament": its leaves represent n players entering the tournament, and each internal node represents a winner of a match played by the players represented by the node's children. Hence, the winner of the tournament is represented by the root of the tree.

3.Define tractable problems.
Problems that can be solved in polynomial time are called tractable.

4.Define intractable problems.
Problems that cannot be solved in polynomial time are called intractable.

5.Define decision tree.
A decision tree is a flowchart-like structure in which internal node represents a "test" on an attribute (e.g. whether a coin flip comes up heads or tails), each branch represents the outcome of the test and each leaf node represents a class label (decision taken after computing all attributes). The paths from root to leaf represent classification rules.
In decision analysis a decision tree and the closely related influence diagram are used as a visual and analytical decision support tool, where the expected values (or expected utility) of competing alternatives are calculated.
A decision tree consists of 3 types of nodes:

- Decision nodes - commonly represented by squares

- Chance nodes - represented by circles
- End nodes - represented by triangles

6.On what basis problems are classified?

Problems are classified into two types based on time complexity. They are

- Polynomial (P) Problem, Non-Polynomial (NP) Problem

7. Define Polynomial (P) problem. (May 17)

Class P is a class of decision problems that can be solved in polynomial time by (deterministic) algorithms. This class of problems is called polynomial.

8. Define Non Polynomial (NP) problem. (Dec 06,May 17)

Class NP is the class of decision problems that can be solved by nondeterministic polynomial algorithms. This class of problems is called nondeterministic polynomial

9. Give the classification of Non Polynomial problem.

Non-Polynomial (NP) problems are classified into two types. They are:

- NP-Hard
- NP-Complete

10.Give some examples of Polynomial problem.

- Selection sort
- Bubble Sort
- String Editing
- Factorial, etc.

11.Give some examples of Non-Polynomial problem.(Dec 06)

- Travelling Salesman Problem
- Knapsack Problem.

12.Define Deterministic algorithm

It has a property that the result of every operation is uniquely defined, and then this type of algorithm is referred to as Deterministic algorithm. Such an algorithm agrees with the way programs are executed on a computer. The machine executing such algorithm is refereed as Deterministic machine.

13.Define Non-Deterministic algorithm.

It has a property that the result of every operation is not uniquely defined, but subject to set of possibilities, then this type of algorithms is referred to as Non-deterministic algorithm.

The possible functions are

- Choice (S)-It arbitrarily chooses one of the value of set S
- Failure ()-signals an Unsuccessful completion
- Success ()-signals an Successful completion

14.Give an example for Class P problem.

- Graph Coloring.

15.Give an example for Class NP problem.

- Traveling Salesman Problem, Knapsack problem

16.Define decision problem.

Any problem for which the answer is either zero or one is called a decision problem. An algorithm for a decision problem is termed a Decision algorithm.

17.Define optimization Problem.
Any problem that involves the identification of an optimal (maximum or minimum) value of a given cost function is known as an optimization problem. An Optimization algorithm is used to solve an optimization problem.

18.When a decision problem is said to be polynomially reducible?
A decision problem D1 is said to be polynomially reducible to a decision problem D2, if there exists a function t that transforms instances of D1 to instances of D2 such that:

- t maps all yes instances of D1 to yes instances of D2 and all no instances of D1 to no instances of D2
- It is computable by a polynomial time algorithm

19.When decision problem D is said to be NP-complete?
A decision problem D is said to be NP-complete if:

- it belongs to class NP, every problem in NP is polynomially reducible to D

20.Define n-queens problem.
The problem is to place n queens on an n × n chessboard so that no two queens attack each other by being in the same row or in the same column or on the same diagonal.

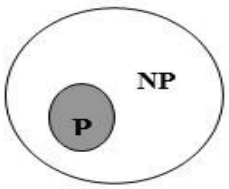
21.Define backtracking. (June 06)
The principal idea is to construct solutions one component at a time and evaluate such partially constructed candidates as follows. If a partially constructed solution can be developed further without violating the problem's constraints, it is done by taking the first remaining legitimate option for the next component. If there is no legitimate option for the next component, no alternatives for any remaining component need to be considered. In this case, the algorithm backtracks to replace the last component of the partially constructed solution with its next option.

22.Define state space tree. (Dec 06,May 16)
It is convenient to implement this kind of processing by constructing a tree of choices being made, called the state-space tree. Its root represents an initial state before the search for a solution begins. The nodes of the first level in the tree represent the choices made for the first component of a solution; the nodes of the second level represent the choices for the second component, and so on.

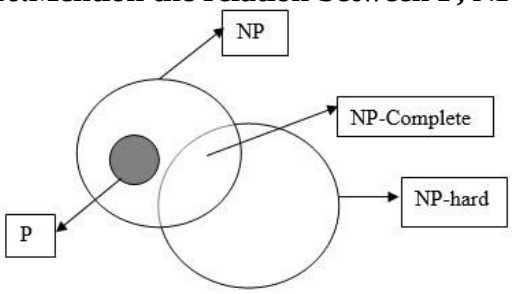
23.When a node in a state space tree is said to promising and non-promising? (Dec 17)
A node in a state-space tree is said to be promising if it corresponds to a partially constructed solution that may still lead to a complete solution; otherwise, it is called non promising. Leaves represent either non promising dead ends or complete solutions found by the algorithm.

24.Define Hamiltonian circuit problem.(June 06, Dec14, May 15)
Let G= (V, E) the € connected graph, with n vertices. A Hamiltonian cycle is a round trip path along n edges of G that visits every vertex once and returns to its starting position.

25.Mention the relation between P and NP.



26.Mention the relation between P, NP, NP-Hard and NP Complete Problem.



27.What is meant by sum of subset problem?

The sum of subset problem is to find a subset of a given set A = {a1, . . . , an} of n positive integers whose sum is equal to a given positive integer d.

28.Define assignment problem.

Assignment problem is the problem of assigning n people to n jobs so that the total cost of the assignment is as small as possible.

29.Define branch and bound method.

- Branch and bound is an algorithm that enhances the idea of generating a state space tree with idea of estimating the best value obtainable from a current node of the decision tree
- If such an estimate is not superior to the best solution seen up to that point in the processing, the node is eliminated from further consideration.

30.Define NP hard problems.

If an NP-hard problem can be solved in polynomial time, then all NP-complete problems can be solved in polynomial time.

31.Define NP complete problems.(Dec 06)

A problem that is NP-complete has the property that it can be solved in polynomial time iff if all other NP-complete problems can also be solved in polynomial time.

32.Define cost of tour. (Dec 14)

Cost of a tour is defined as the sum of the cost of the edges on the tour.

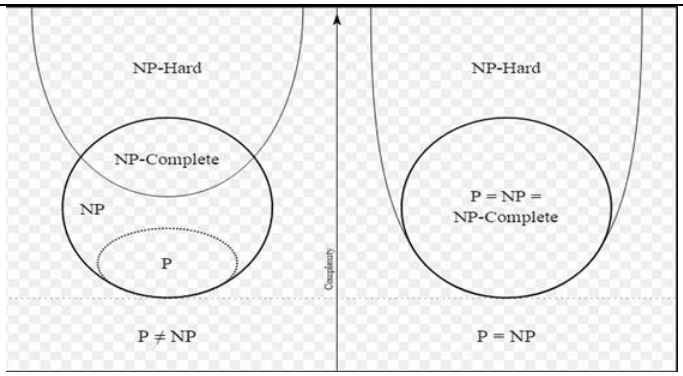
33.Define Live and Dead nodes. (Dec 14)

- Live node a node which has been generated and all of whose children have not yet been generated.
- E-node the live node whose children are currently being generated.
- Dead node a generated node which is not to be expanded further or all of whose children have been generated.

34.How NP-hard problems are different from NP-Complete? (May 15)

NP-hard : If an NP-hard problem can be solved in polynomial time, then all NP-complete problems can be solved in polynomial time.

NP-Complete: A problem that is NP-complete has the property that it can be solved in polynomial time iff if all other NP-complete problems can also be solved in polynomial time.



35.Give the purpose of lower bound. (May 16)

Given a class of algorithms for solving a particular problem, a lower bound indicates the best possible efficiency any algorithm from this class can have.

36.What is Euclidean minimum spanning tree problem? (May 16)

The Euclidean minimum spanning tree or EMST is a minimum spanning tree of a set of n points in the plane, where the weight of the edge between each pair of points is the Euclidean distance between those two points.

37.What is an articulation point in a graph? (May 17)

A vertex of a connected graph is said to be its articulation point if its removal with all edges incident to it breaks the graph into disjoint pieces.

38.Define NP completeness and NP hard.

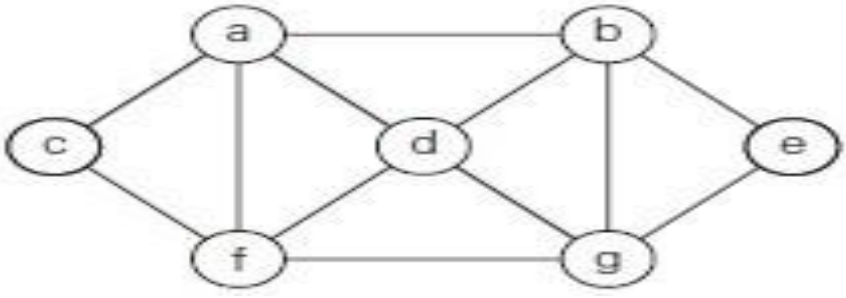
A decision problem D is said to be NP-complete if:

- it belongs to class NP
- every problem in NP is polynomially reducible to D

NP-hard problems: problems that are at least as hard as NP-complete problems (it follows the property 2 mentioned above and doesn't need to follow property 1)

Part B & C- (16 marks & 8 marks)

1. What is backtracking? Write the template of general backtracking algorithm and explain in detail with suitable example. (June 07)
2. Explain n-queen's problem. Draw a portion of the state space tree and perform backtracking search for a solution to 4-queens problem. (June 06,Dec 07,14)
3. Explain how backtracking method is applied to solve subset sum problem with suitable example.(June 0,Dec 07)
4. Write a pseudo code for backtracking algorithm and apply backtracking to solve the following instances of the subset sum problem: S= {1, 3, 4,5} d=11 and d=8. (Dec 06)
5. Explain in detail about Hamiltonian circuit problem with suitable example.



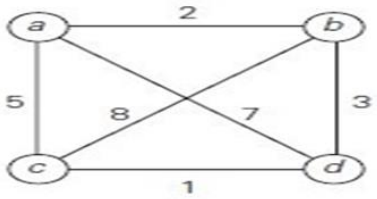
6.Explain assignment problem using branch and bound method or Explain how job assignment problem could be solved, given n tasks and n agents where each agent has a cost to complete each task, using Branch and

bound technique.

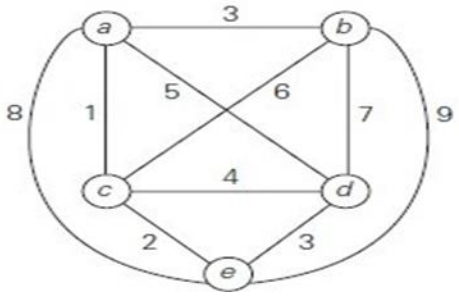
7. Solve the following assignment problem using branch and bound technique. Explain in detail how branch and bound technique is useful for solving assignment problems.

	job 1	job 2	job 3	job 4	
$C =$	9	2	7	8	person *a*
	6	4	3	7	person *b*
	5	8	1	8	person *c*
	7	6	9	4	person *d*

8.Apply Branch and Bound algorithm to solve the Travelling salesman problem for (May 17)



9.Solve the following instance of the travelling salesman problem by branch and bound method and explain in detail. (Dec 07)



10. Apply the branch and bound algorithm to solve the following knapsack problem and explain in detail.

item	weight	value
1	10	$100
2	7	$63
3	8	$56
4	4	$12

The knapsack capacity W is 16.

11. Explain knapsack problem, LIFOSearch and FIFO search using branch and bound technique. (Dec 07)
12.Explain travelling salesman problem using branch and bound technique.
13.Explain the concept of P, NP, NP hard and NP complete.
14.Explain in detail about approximation algorithms for NP hard problems. How all you quantify the accuracy of an approximation algorithm? (May 17)
15.Discuss about Lower bound arguments?

Faculty Incharge
(Mrs.C.Sangeetha)

Head of the Department
(Dr.A.Punitha)