INSTITUTE OF AERONAUTICAL ENGINEERING

(Autonomous)

Dundigal, Hyderabad -500 043

INFORMATION TECHNOLOGY

TUTORIAL QUESTION BANK

Course Title	**DESIGN AND ANALYSIS OF ALGORITHMS**				
Course Code	AITB05				
Programme	B.Tech				
Semester	IV	CSE | IT			
Course Type	Core				
Regulation	**IARE - R18**				
Course Structure	**Theory**			**Practical**	
	Lectures	**Tutorials**	**Credits**	**Laboratory**	**Credits**
	3	1	4	3	1.5
Chief Coordinator	Ms. G Geetha,Assistant Professor				
Course Faculty	Dr. M Purushotham Reddy, AssociateProfessor Dr.K Suvarchala, AssociateProfessor Ms.E Uma shankari, Assistant Professor Ms.G Srileka, Assistant Professor				

COURSE OVERVIEW:

The primary objective of this course is to introduce the concept of algorithm as a precise mathematical concept, and study how to design algorithms, establish their correctness, study their efficiency and memory needs. The course consists of a strong mathematical component in addition to the design of various algorithms.

COURSE OBJECTIVES:

The course should enable the students to:

I	Assess how the choice of data structures and algorithm design methods impacts the performance of programs.
II	Solve problems using data structures such as binary search trees, and graphs and writing programs for these solutions.
III	Choose the appropriate data structure and algorithm design method for a specified application
IV	Solve problems using algorithm design methods such as the greedy method, divide and conquer, dynamic programming, backtracking, and branch and bound and writing programs for these solutions

COURSE OUTCOMES (COs):

COs	Course Outcome
CO 1	Understand the concept of pseudo code for writing an algorithm and acquire ability to analyze the asymptotic performance of various algorithms
CO 2	Explore the concept of trees and graphs and get familiarity of analysis of various graphs, tree traversal algorithms.
CO 3	Understand algorithm designing techniques such as Greedy approach Dynamic programming and explore to various related application problems.
CO 4	Synthesize efficient algorithm design paradigms back tracking, Branch & Bound in solving common analytical problems.
CO 5	Understand the variations among tractable and intractable problems and able to classify P and NP classes.

COURSE LEARNING OUTCOMES:

Students, who complete the course, will have demonstrated the ability to do the following:

AITB05.01	Describe Pseudo code for expressing algorithms.
AITB05.02	Summarize the concept of Space complexity, time complexity.
AITB05.03	Describe Big O notation, omega notation, theta notation, little o notation and amortized complexity.
AITB05.04	Use the concept of Divide and Conquer such as general method, binary search, quick sort.
AITB05.05	Describe the concept of merge sort, Strassen's matrix multiplication.
AITB05.06	Determine disjoint set operations, union and find algorithms.
AITB05.07	Understand efficient non recursive binary tree traversal algorithms.
AITB05.08	Describe the concept of spanning trees with suitable examples.
AITB05.09	Use breadth first search and depth first search graph traversals.
AITB05.10	Describe connected components, biconnected components.
AITB05.11	Understand general method of greedy method, job sequencing with deadlines, knapsack problem.
AITB05.12	Analyze the concept of minimum cost spanning trees, single source shortest paths.
AITB05.13	Describe general method of dynamic programming, matrix chain multiplication.
AITB05.14	Understand optimal binary search trees, 0/1 knapsack problem, single source shortest paths.
AITB05.15	Define all pairs shortest paths problem, the travelling salesperson problem.
AITB05.16	Discuss the concept of Backtracking, the 8 queens problem.
AITB05.17	Understand sum of subsets problem, graph coloring.
AITB05.18	Summarize the concept of Hamiltonian cycles, Branch and bound.
AITB05.19	Discuss 0/1 knapsack problem, least cost branch and bound solution.
AITB05.20	Apply the concept of first in first out branch and bound solution, travelling salesperson problem.
AITB05.21	Knowledge about basic concepts of NP Hard and NP Complete, Non-deterministic algorithms.
AITB05.22	Apply Working with the classes NP - Hard and NP.
AITB05.23	Understand NP Hard problems, clique decision problem.
AITB05.24	Implement chromatic number decision problem.
AITB05.25	Discuss Cook's theorem in NP Hard and NP Complete problems.

TUTORIAL QUESTION BANK

MODULE- I

INTRODUCTION

PART – A (SHORT ANSWER QUESTIONS)

S. No	Questions	Blooms Taxonomy Level	Course Outcomes	Course Learning Outcomes
1.	List out various basic characteristics of an algorithm.	Remember	CO1	AITB05.01
2.	Illustrate the importance of probability analysis?	Remember	CO1	AITB05.02
3.	State the best case and worst case analysis for linear search	Remember	CO1	AITB05.02
4.	Construct recurrence equation for the worst case behavior of a merge sort?	Understand	CO1	AITB05.04
5.	Analyze the average case time complexity of quick sort?	Understand	CO1	AITB05.04
6.	List best case, average case and worst case efficiency of a binary search algorithm?	Remember	CO1	AITB05.02
7.	Explore the term amortized Complexity.	Remember	CO1	AITB05.03
8.	Describe the role of space complexity and time complexity of an Algorithm.	Remember	CO1	AITB05.02
9.	Illustrate the control abstraction of divide and conquer.	Understand	CO1	AITB05.04
10.	List out any two drawbacks of binary search algorithm.	Understand	CO1	AITB05.04

PART – B (LONG ANSWER QUESTIONS)

S. No	Questions	Blooms Taxonomy Level	Course Outcomes	Course Learning Outcomes
1.	Describe various asymptotic notations with suitable examples and examine the best case, average case and worst case analysis of linear search algorithm.	Remember	CO1	AITB05.03
2.	Differentiate between priori analysis and posteriori analysis. Illustrate the same with an example	Understand	CO1	AITB05.01
3.	Demonstrate Divide and Conquer technique through binary search algorithm and analyze its time complexity for all three cases	Understand	CO1	AITB05.04
4.	Elucidate Quick sort algorithm and simulate it for the following data: 20, 35, 10, 16, 54, 21, 25	Understand	CO1	AITB05.04
5.	Design and demonstrate iterative binary search algorithm and Examine its Time complexity in Worst case, Best case and Average cases.	Remember	CO1	AITB05.04
6.	Illustrate merge sort algorithm and discuss time complexity in both worst case and average cases.	Understand	CO1	AITB05.05
7.	Describe the advantage of Strassen's matrix multiplication when compared to normal matrix multiplication for the any two 8 x 8 matrices.	Understand	CO1	AITB05.05
8.	Describe amortized analysis and Examine how amortized complexity, actual complexity related.	Understand	CO1	AITB05.03
9.	Demonstrate Randomized algorithm. Describe the procedure to analyze Randomized algorithms.	Remember	CO1	AITB05.01
10.	Construct the Merge sort algorithm for the list of numbers: 78, 32, 42, 62, 98, 12, 34, 83	Understand	CO1	AITB05.05

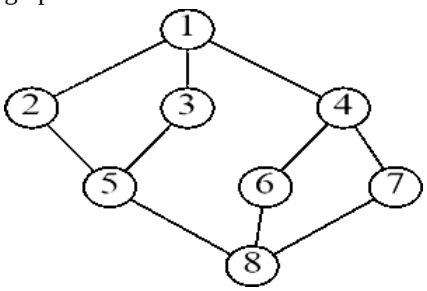
11.	Solve the recurrence relation using substitution method T(n)= { T(1) n=1 aT(n/b)+f(n) n>1 ,where a=5,b=4,and f(n)=cn^2	Understand	CO1	AITB05.04
12.	Describe the Pseudo code conventions for specifying algorithms of recursive and an iterative algorithm to compute n!	Remember	CO1	AITB05.01

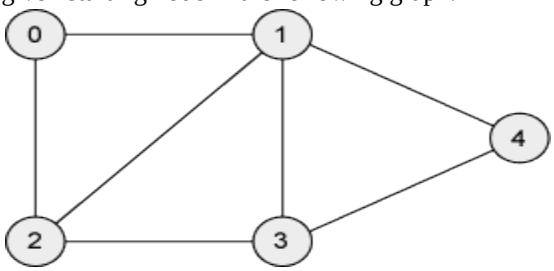
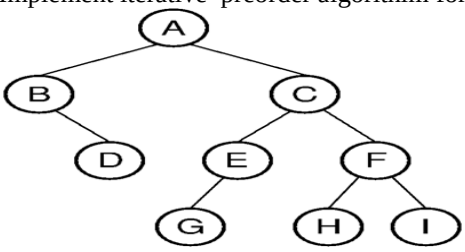
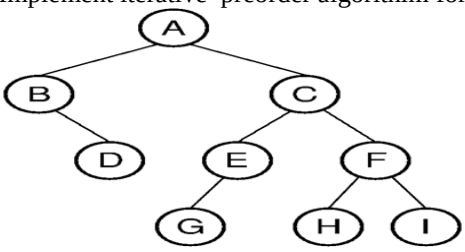
13.	Determine the frequency counts for all statements in the following algorithm segment.
	i=1;
	while(i<=n) do
	{
	x=x+1;
	i=i+1;
	}
		Understand	CO1	AITB05.05

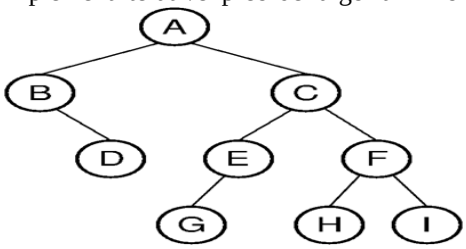
14.	Write about stable sorting method? Is merge sort a stable sorting algorithm? Justify your answer.	Remember	CO1	AITB05.05
15.	Describe the recurrence relation with an example. Illustrate any two techniques to solve recurrence relation.	Remember	CO1	AITB05.04
16.	Develop repetitive and recursive algorithms for printing first n Fibonacci numbers and analyze both algorithms.	Understand	CO1	AITB05.04
17.	If $f(n)=5n^2 + 6n + 4$, then prove that f(n) is $O(n^2)$	Remember	CO1	AITB05.04

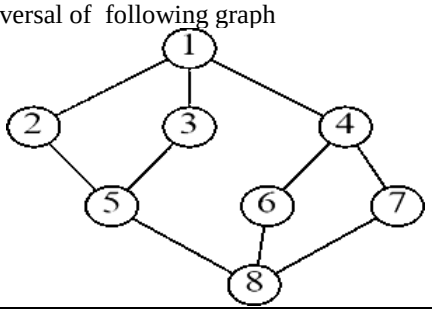
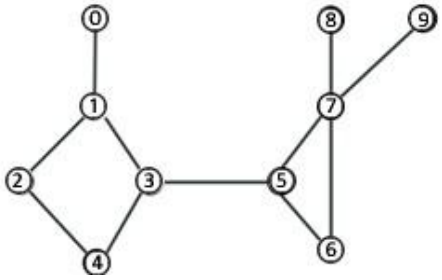
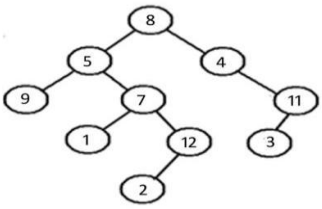
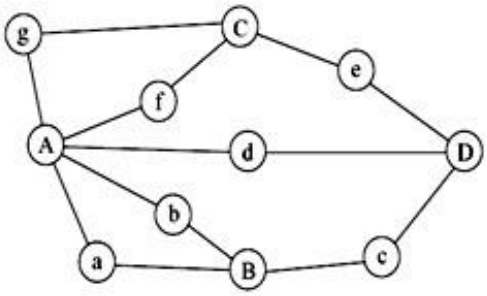
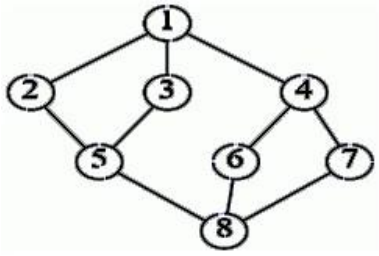
PART – C (PROBLEM SOLVING AND CRITICAL THINKING QUESTIONS)

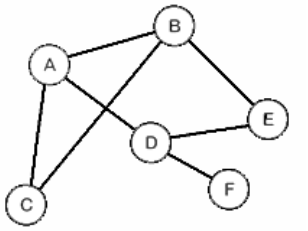
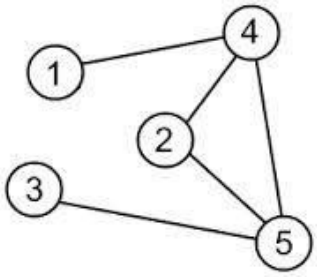
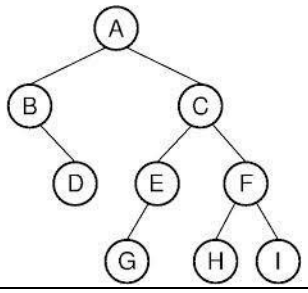
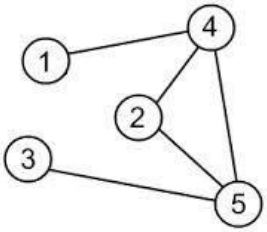
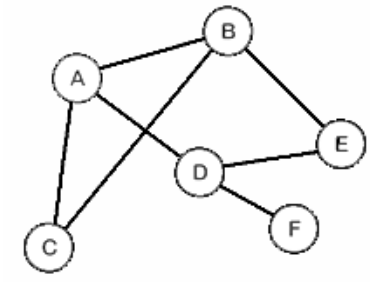
S. No	Questions	Blooms Taxonomy Level	Course Outcomes	Course Learning Outcomes
1.	Solve the following recurrence relation T(n)=2 T(n/2) + n, and T(1)=2	Understand	CO1	AITB05.04
2.	Solve the following recurrence relation $T(n) = 7T(n/2)+cn^2$	Understand	CO1	AITB05.04
3.	Solve the recurrence relation T(n)=1 , n=1 T(n)=T(n/2) + c, n>1 and n is a power of 2	Understand	CO1	AITB05.04
4.	Construct the quick sort algorithm and simulate it for following data sequence: 3 5 9 7 1 4 6 8 2	Understand	CO1	AITB05.04
5.	Construct the various tracing steps of merge sort and quicksort and analyze the time complexity for the following data: 33, 44, 2, 10, 25, 79, 86, 47, 14, 36	Understand	CO1	AITB05.05
6.	Derive the average and best case time complexity of quick sort and merge sort algorithm.	Remember	CO1	AITB05.04
7.	Construct merge sort alrotighm on following letters H, K, P,C,S,K,R,A,B,L and analyse its best, average and worst cases time complexity.	Understand	CO1	AITB05.05
8.	Examine when Strassen's method outperforms the traditional matrix multiplication method. How many numbers of multiplication operations are required during multiplication of two matrices with size of 8 x 8 in Stressen's method?	Remember	CO1	AITB05.05
9.	Construct and solve recurrence relation for Strassen's matrix multiplication.	Understand	CO1	AITB05.05

10.	Solve the following recurrence relation T(n)=2 T(n/2) + 1, and T(1)=2	Understand	CO1	AITB05.04
	MODULE- II			
	SEARCHING AND TRAVERSAL TECHNIQUES			
	PART – A (SHORT ANSWER QUESTIONS)			
S. No	**Questions**	**Blooms Taxonomy Level**	**Course Outcomes**	**Course Learning Outcomes**
1.	List out various operation on disjoinsets.	Remember	CO2	AITB05.06
2.	Define spanning tree and minimal spanning tree	Remember	CO2	AITB05.08
3.	Analyse time complexitie of depth first search for the inputs of adjacency list and adjacency matrix.	Remember	CO2	AITB05.09
4.	Evaluate time complexities of breadth first search for the inputs of adjacency list and adjacency matrix	Understand	CO2	AITB05.09
5.	Distinguish between breadth first search and depth first search	Understand	CO2	AITB05.09
6.	List out various binary tree traversal techniques.	Remember	CO2	AITB05.07
7.	State what is an articulation point.	Understand	CO2	AITB05.10
8.	Distinguish between connected and disconnected graphs	Remember	CO2	AITB05.10
9.	List out different data structures used for implementing the breadth first search and depth first search	Remember	CO2	AITB05.09
10.	Define bi-connected component.	Remember	CO2	AITB05.10
	PART – B (LONGANSWER QUESTIONS)			
S .No	**Questions**	**Blooms Taxonomy Level**	**Course Outcomes**	**Course Learning Outcomes**
1.	Elucidate weighting rule for finding UNION of sets and collapsing rule.	Understand	CO2	AITB05.06
2.	Simulate the behavior of weighted union on the following for some sequence of unions starting from the initial configuration p[i]=-count[i]=-1,≤ i ≤ 8= n Union(1,2),Union(3,4),Union(5,6),Union(7,8),Union(1,3), Union(5,7),Union(1,5)	Understand	CO2	AITB05.06
3.	Demonstrate breadth first traversal algorithm for the following graph  With the starting vertices as 6 and 7.	Understand	CO2	AITB05.08
4.	Design and analyze iterative versions of inorder binary tree traversal algorithms .	Remember	CO2	AITB05.07

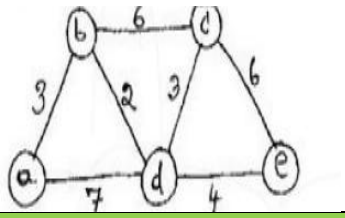
5.	Design and analyze iterative versions of preorder binary tree traversal algorithms.	Remember	CO2	AITB05.07
6.	Design and analyze iterative versions of post order binary tree traversal algorithms.	Remember	CO2	AITB05.07
7.	Compare the approaches of BFS and DFS methods and derive the time complexities of both methods for the inputs of adjacency lists and adjacency matrix separately.	Remember	CO2	AITB05.09
8.	Describe construction of spanning tree through dft algorithms with an example.	Remember	CO2	AITB05.08
9.	Demonstrate the construction of a binary tree from given inorder and preorder traversals.	Understand	CO2	AITB05.07
10.	Describe construction of spanning tree through bft algorithms	Understand	CO2	AITB05.08
11.	Illustrate the procedure to identify that the given graph is connected or not	Understand	CO2	AITB05.09
12.	Elucidate the concept of biconnected component with an example	Remember	CO2	AITB05.10
13.	Simulate bft algorithm to print all the nodes reachable from a given starting node in the following graph. 	Remember	CO2	AITB05.09
14.	Implement iterative preorder algorithim for the following tree 	Understand	CO2	AITB05.07
15.	Implement iterative preorder algorithim for the following tree 	Understand	CO2	AITB05.07

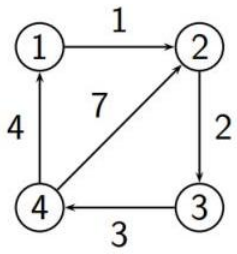
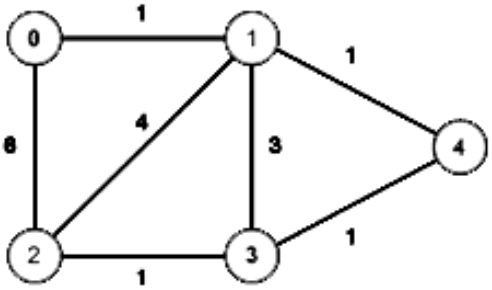
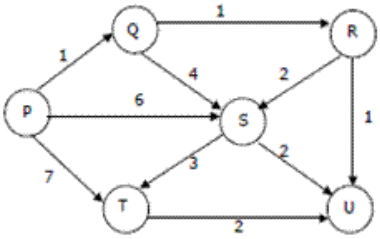
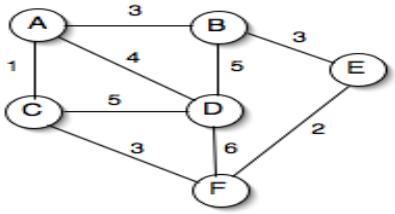
16.	Implement iterative preorder algorithim for the following tree 	Understand	CO2	AITB05.07
17.	Construct binary tree from the following In order sequence: D B E A F C and Preorder sequence: A B D E C F	Remember	CO2	AITB05.07
18.	Interpret the advantages of collapse find over find algorithm with an example.	Understand	CO2	AITB05.06
19.	Construct binary tree from the following inorder sequence: 4, 8, 2, 5, 1, 6, 3, 7 and postorder sequence: 8, 4, 5, 2, 6, 7, 3, 1	Understand	CO2	AITB05.07
20.	Check whether a given graph is connected or not using DFS method.	Understand	CO3	AITB05.10

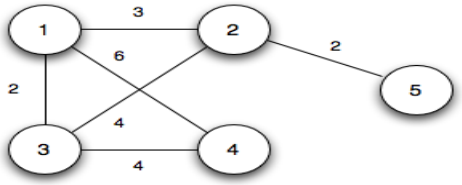
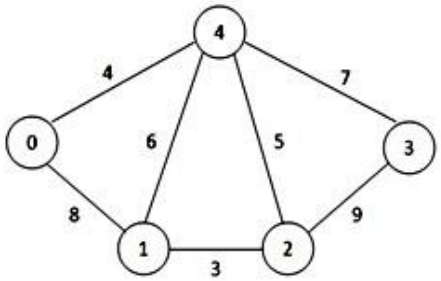
PART – C (PROBLEM SOLVING AND CRITICAL THINKING QUESTIONS)				
1.	Illustrate BFS traversal of following graph 	Understand	CO2	AITB05.09
2.	List out the articulation points from the following given graph 	Remember	CO2	AITB05.10
3.	Analyze in order, pre order, post order traversal of the following tree 	Remember	CO2	AITB05.07
4.	Illustrate DFS and BFS traversals of following graph 	Understand	CO2	AITB05.09
5.	Illustrate DFS traversal of following graph 	Understand	CO2	AITB05.09

6.	Illustrate BFS traversal of the following graph 	Understand	CO2	AITB05.09
7.	List the articulation points from the following graph 	Understand	CO2	AITB05.10
8.	Simulate inorder, preorder, post order traversal of the following tree 	Understand	CO2	AITB05.07
9.	Illustrate BFS and DFS traversals of following graph 	Understand	CO2	AITB05.09
10.	Illustrate DFS traversal of following graph 	Understand	CO2	AITB05.09

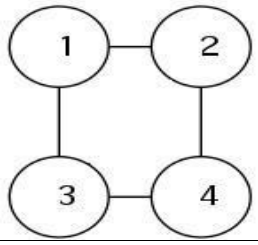
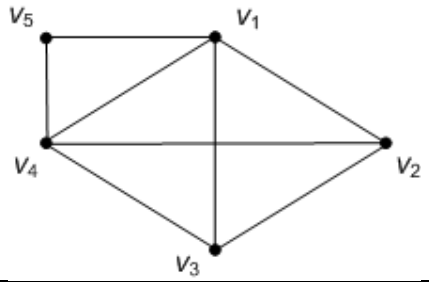
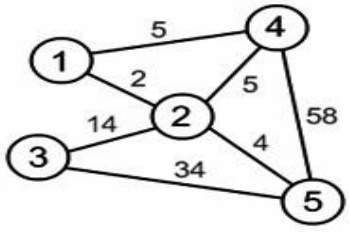
	MODULE- III			
	GREEDY METHOD AND DYNAMIC PROGRAMMING			
	PART – A (SHORT ANSWER QUESTIONS)			
S. No	**Questions**	**Blooms Taxonomy Level**	**Course Outcomes**	**Course Learning Outcomes**
1.	Define greedy method	Remember	CO3	AITB05.11
2.	Define job sequencing with deadlines problem	Remember	CO3	AITB05.11
3.	Write about minimum cost spanning tree	Remember	CO3	AITB05.12
4.	State single source shortest path problem	Understand	CO3	AITB05.11
5.	What is feasible solution.	Remember	CO3	AITB05.11
6.	Write about optimal solution.	Remember	CO3	AITB05.11
7.	State the time complexities of prims and kruskals algorithms	Understand	CO3	AITB05.12
8.	Define knapsack problem.	Remember	CO3	AITB05.11
9.	Analyze the time complexities of Fractional Knap Sack problem	Understand	CO3	AITB05.12
10.	List out the applications of greedy method.	Remember	CO3	AITB05.11
1.	Define dynamic programming.	Remember	CO3	AITB05.13
2.	State the principle of optimality	Understand	CO3	AITB05.13
3.	List out the features of dynamic programming	Remember	CO3	AITB05.13
4.	Distinguish between greedy method from dynamic programming	Understand	CO3	AITB05.13
5.	State the formula for computing cost of binary search tree.	Understand	CO3	AITB05.14
6.	State the time complexity of travelling salesperson problem using dynamic programming.	Understand	CO3	AITB05.1
7.	Define dominance rule.	Understand	CO3	AITB05.14
8.	State the time complexity of all pairs shortest paths problem.	Understand	CO3	AITB05.15
9.	Define 0/1 knapsack problem	Remember	CO3	AITB05.14
10.	What is matrix chain multiplication problem?	Remember	CO3	AITB05.13
	PART – B (LONG ANSWER QUESTIONS)			
1.	With the algorithm, describe job sequencing with deadlines problem.	Remember	CO3	AITB05.11
2.	Elucidate single source shortest path problem by using greedy method	Understand	CO3	AITB05.12
3.	Demonstrate briefly about the knapsack problem with suitable example	Understand	CO3	AITB05.11
4.	State and elucidate an algorithm for Prim's algorithm.	Understand	CO3	AITB05.12
5.	Elucidate kruskals algorithm with suitable example	Understand	CO3	AITB05.12
6.	Prove that Prim's method generates minimum-cost spanning tree.	Understand	CO3	AITB05.12
7.	State control abstraction of greedy method and explain how it is useful for real time problems	Understand	CO3	AITB05.11
8.	Describe the greedy method for generating the shortest paths	Understand	CO3	AITB05.12
9.	Estimate the time complexities of Prim's and Kruskal's algorithms.	Understand	CO3	AITB05.12
10.	Distinguish between the Prim's algorithm and Kruskal's algorithm.	Remember	CO3	AITB05.12

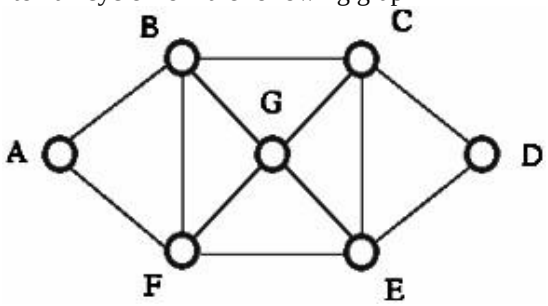
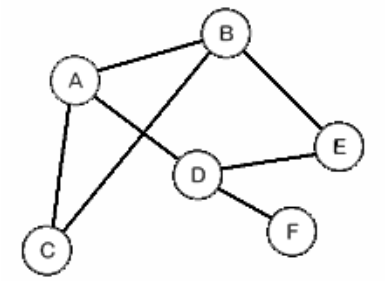
11.	Obtain the minimum cost spanning tree for a graph G(6,10) with vertices named as a, b, c, d, e, f and edges ab=3, bc=1, af=5, ae=6, ed=8, fe=2, fd=5, cd=6, cf=4 and bf=4 using Prim's algorithm and showing results in each stages.	Understand	CO3	AITB05.12
12.	Design the control abstraction for subset paradigm using greedy method. Solve the job sequencing with deadline problem using greedy method for the given data N=7,P={3,5,20,18,1,6,30} are profits and D={1,3,4,3,5,1,2} are deadline respectively.	Understand	CO3	AITB05.11
13.	Obtain the minimum cost spanning tree for a graph G(6,10) with vertices named as a,b,c,d,e,f and edges ab=1, bc=3, af=9,ae=4, ed=6, fe=4, fd=5, cd=6, cf=4 and bf=4 using Kruskal's algorithm and showing results in each stages.	Understand	CO3	AITB05.12
14.	Obtain the shortest path from source a to all other vertices in the graph shown in below Fig. Using greedy method .Give the greedy criterion used. 	Understand	CO3	AITB05.12
1.	Elucidate optimal binary search tree algorithm with example	Understand	CO3	AITB05.14
2.	Elucidate 0/1 knapsack problem with an example	Understand	CO3	AITB05.14
3.	Narrate all pairs shortest path problem with an example	Understand	CO3	AITB05.14
4.	Describe the travelling salesman problem and discuss how to solve it using dynamic programming?	Understand	CO3	AITB05.15
5.	Illustrate matrix chain multiplication with example	Understand	CO3	AITB05.13
6.	Describe single source shortest path problem with example using dynamic programming	Remember	CO3	AITB05.14
7.	Describe the principle of optimality in Dynamic Programming with example.	Remember	CO3	AITB05.13
8.	Analyze the time complexity of optimal binary search tree.	Understand	CO3	AITB05.14
9.	Illustrate Informal knapsack algorithm by using dynamic Programming	Remember	CO3	AITB05.14
10.	State the recursive steps of optimal binary search tree with an example	Understand	CO3	AITB05.14
11.	List out all possible binary search trees with keys 10, 12, 20 and find optimal binary search tree.	Understand	CO3	AITB05.14
12.	State the recurrence used to solve knapsack problem using dynamic programming and explain in brief the same. Solve the following Knapsack problem using dynamic programming. Capacity W=5 Item 1 2 3 4 Weight 2 1 3 2 Value 12 10 20 15	Understand	CO3	AITB05.14
13.	Solve the 0/1 knapsack problem using dynamic programming. Given profits (60,100,120), weights (10, 20, 30) and capacity 50.	Understand	CO3	AITB05.14

14.	Obtain shortest distances using all pairs shortest path algorithm 	Understand	CO3	AITB05.15
PART – C (PROBLEM SOLVING AND CRITICAL THINKING QUESTIONS)				
1.	Obtain the optimal solution for job sequencing with deadlines using greedy method. N=4, profits (p1,p2,p3,p4) = (100,10,15,27), Deadlines (d1,d2,d3,d4) = (2,1,2,1)	Understand	CO3	AITB05.11
2.	Obtain the optimal solution for knapsack problem using greedy method N=3, M= 20, (p1,p2,p3)= (25,24,15), (w1,w2,w3) =(18,15,10)	Understand	CO3	AITB05.11
3.	Obtain the minimum cost spanning tree using Prims algorithm 	Understand	CO3	AITB05.12
4.	Solve single source shortest path algorithm for the following graph 	Understand	CO3	AITB05.12
5.	Obtain Minimum Cost Spanning Tree of a given undirected graph using Kruskal's algorithm. 	Understand	CO3	AITB05.12

6.	Obtain Minimum Cost Spanning Tree of a given undirected graph using Prim's algorithm.  	Understand	CO3	AITB05.12
7.	Obtain the optimal solution when n=5, (p1, p2,...) = (20,15, 10,5,1) and (d1,d2,...)=(2,2,1,3,3).	Understand	CO3	AITB05.11
1.	Obtain wij, cij, rij, 0<=i<=j<=4,p1=1/10, p2=1/5, p3=1/10, p4=1/120, q0=1/5, q1=1/10, q2=1/5, q3=1/20,q4=1/20 by using optimal binary search tree algorithm.	Understand	CO3	AITB05.14
2.	Construct optimal binary search for (a1, a2, a3, a4) = (do, if,int, while), p(1 : 4) = (3,3,1,1) q(0 : 4)= (2,3,1,1,1)	Understand	CO3	AITB05.14
3.	Solve the solution for 0/1 knapsack problem using dynamic programming(p1,p2,p3, p4) = (11, 21, 31, 33), (w1, w2, w3, w4) = (2, 11, 22, 15), M=40, n=4	Understand	CO3	AITB05.14
4.	Solve the solution for 0/1 knapsack problem using dynamic programming N=3 , m=6 profits (p1,p2,p3) = (1,2,5) weights (w1,w2,w3) = (2,3,4)	Understand	CO3	AITB05.14
5.	Calculate shortest distance by using all pairs shortest path algorithm  	Understand	CO3	AITB05.15
6.	Obtain the shortest tour of traveling sales person for the following cost matrix using dynamic Programming $\begin{bmatrix} \infty & 12 & 5 & 7 \\ 11 & \infty & 13 & 6 \\ 4 & 9 & \infty & 18 \\ 10 & 3 & 2 & \infty \end{bmatrix}$	Understand	CO3	AITB05.14
7.	Compute the minimum number of multiplications required to multiply chain of matrices A1,A2,A3,A4 of order 10x5, 5x10, 10x5, 5x20	Understand	CO3	AITB05.13
8.	Obtain the solution to knapsack problem by Dynamic Programming method n=6, (p1, p2,...p6)=(w1,w2,...w6)=(100,50,20,10,7,3) and m=165.	Understand	CO3	AITB05.14

	MODULE- IV			
BACKTRACKING AND BRANCH AND BOUND				
PART – A (SHORT ANSWER QUESTIONS)				
1.	State the principle of Backtracking	Understand	CO4	AITB05.16
2.	Write control abstraction for backtracking	Understand	CO4	AITB05.16
3.	List out the applications of backtracking?	Remember	CO4	AITB05.16
4.	Define a dead node	Remember	CO4	AITB05.16
5.	Distinguish between live node and dead node	Understand	CO4	AITB05.16
6.	Write briefly about state space tree	Remember	CO4	AITB05.16
7.	State 8 – Queens problem	Understand	CO4	AITB05.16
8.	State Sum of Subsets problem	Understand	CO4	AITB05.17
9.	Illustrate briefly about E-node	Remember	CO4	AITB05.18
10.	What is D-search?	Remember	CO4	AITB05.18
PART – B (LONG ANSWER QUESTIONS)				
1.	Write an algorithm for N-queens problem using backtracking?	Understand	CO4	AITB05.16
2.	State subset-sum problem and discuss the possible solution strategies using backtracking.	Remember	CO4	AITB05.17
3.	Describe graph coloring problem and write an algorithm for m-coloring problem?	Understand	CO4	AITB05.17
4.	Write an algorithm for Hamiltonian cycle with an example?	Understand	CO4	AITB05.18
5.	List out the properties of Least Cost search?	Remember	CO4	AITB05.19
6.	Describe the control abstraction for Least Cost Search?	Remember	CO4	AITB05.19
7.	State the principle of FIFO branch and bound.	Remember	CO4	AITB05.20
8.	Elucidate the principle of LIFO branch and bound	Remember	CO4	AITB05.20
9.	Elucidate the method of reduction to solve travelling sales person problem using branch and bound technique.	Remember	CO4	AITB05.20
10.	Illustrate TSP using branch and bound method with example?	Understand	CO4	AITB05.20
11.	Explain the basic principle of Backtracking and list the applications of Backtracking.	Understand	CO4	AITB05.16
12.	Solve the following instance for the subset problem using backtracking technique s=(1,3,4,5) and d=11.	Understand	CO4	AITB05.16
13.	Draw the portion of the state space tree generated by LCBB for the knapsack instance: n=5, (p1,p2,p3,p4,p5)=(w1,w2,w3,w4,w5)=(4,4,5,8,9), and m=15.	Understand	CO4	AITB05.19
14.	Write an algorithm for 4-queens problem by using backtracking	Understand	CO4	AITB05.16
15.	Solve the following instance for the subset problem using backtracking technique s= (6,5,3,7) and d=15.	Understand	CO4	AITB05.17
PART – C (PROBLEM SOLVING AND CRITICAL THINKING QUESTIONS)				
1.	Construct the state space tree degenerated by 4 queens problem	Understand	CO4	AITB05.16
2.	Solve the following instance of the sum of subsets problem S={5,10,12,13,15,18} and d=30 by using backtracking algorithm .	Understand	CO4	AITB05.16

3.	Design the state space tree that generated all possible 3-color,4-node graph 	Understand	CO4	AITB05.16
4.	Identify Hamiltonian cycle from the following given graph 	Understand	CO4	AITB05.18
5.	Solve the following instance of travelling sales person problem using Least Cost Branch and Bound $\begin{bmatrix} \infty & 12 & 5 & 7 \\ 11 & \infty & 13 & 6 \\ 4 & 9 & \infty & 18 \\ 10 & 3 & 2 & \infty \end{bmatrix}$	Understand	CO4	AITB05.20
6.	Construct the portion of state space tree generated by LCBB by the following knapsack problem n=5 , (p1,p2,p3,p4,p5) =(10,15,6, 8, 4), (w1,w2,w3,w4,w5)=(4,6,3,4,2) and m=12	Understand	CO4	AITB05.19
7.	Draw the portion of state space tree generated by FIFO knapsack for the instance N=4 , (P1, P2, P3, P4)= (10, 10, 12, 18) , (w1, w2,w3,w4) = (2, 4, 6, 9) , m=15	Understand	CO4	AITB05.20
8.	Solve the following instance of travelling sales person problem using Least Cost Branch Bound 	Understand	CO4	AITB05.20

9.	Obtain Hamiltonian cycle from the following graph 	Understand	CO4	AITB05.18
10.	Describe the backtracking algorithm to color the following graph 	Understand	CO4	AITB05.17

MODULE- IV

NP-HARD AND NP-COMPLETE PROBLEMS

PART – A (SHORT ANSWER QUESTIONS)

S. No	Question	Blooms Taxonomy Level	Course Outcomes	Course Learning Outcomes
1.	Elucidate class P	Remember	CO5	AITB05.22
2.	Compare NP-hard and NP-completeness	Understand	CO5	AITB05.21
3.	State NP- hard problem	Remember	CO5	AITB05.23
4.	Describe NP-complete problem	Remember	CO5	AITB05.21
5.	State Deterministic problem?	Remember	CO5	AITB05.21
6.	Elucidate Non-deterministic problem	Remember	CO5	AITB05.21
7.	Define a decision problem?	Remember	CO5	AITB05.23
8.	Narrate Maxclique problem?	Remember	CO5	AITB05.23
9.	State Halting problem	Remember	CO5	AITB05.23
10.	Describe vertex cover problem.	Remember	CO5	AITB05.24

PART – B (LONG ANSWER QUESTIONS)

1.	State and prove Cook's theorem	Understand	CO5	AITB05.25
2.	Describe deterministic and non-deterministic algorithms	Remember	CO5	AITB05.21
3.	Develop non deterministic algorithm for sorting and searching	Understand	CO5	AITB05.21
4.	Design non-deterministic knapsack algorithm	Understand	CO5	AITB05.21
5.	Elucidate how P and NP problems are related	Remember	CO5	AITB05.23
6.	Distinguish between NP- hard and NP-complete problems	Understand	CO5	AITB05.21
7.	Narrate decision problem with an example	Understand	CO5	AITB05.24

8.	Describe chromatic number decision problem and clique decision problem	Remember	CO5	AITB05.23
9.	Explore the strategy to prove that a problem is NP-hard	Remember	CO5	AITB05.23
10.	Describe intractable problems with examples	Remember	CO5	AITB05.23
PART – C (PROBLEM SOLVING AND CRITICAL THINKING QUESTIONS)				
1.	Prove that satisfiability is at most three literals reduces to chromatic number	Understand	CO5	AITB05.24
2.	Prove Hamiltonian cycle is in NP	Understand	CO5	AITB05.24
3.	Prove circuit-SAT is in NP	Understand	CO5	AITB05.21
4.	List out two problems that have polynomial time algorithms justify your answer	Understand	CO5	AITB05.21
5.	Demonstrate 3CNF satisfiability problem	Remember	CO5	AITB05.21
6.	Demonstrate P type problems with examples	Remember	CO5	AITB05.21

Prepared By:
G Geetha, Assistant Professor

HOD, CSE