# INSTITUTE OF AERONAUTICAL ENGINEERING

**(Autonomous)**

Dundigal, Hyderabad -500 043

## COMPUTER SCIENCE AND ENGINEERING

## ASSIGNMENT– I AND II QUESTIONS

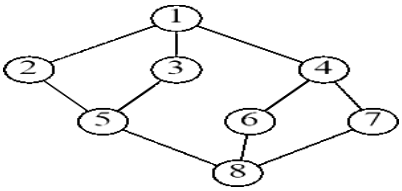
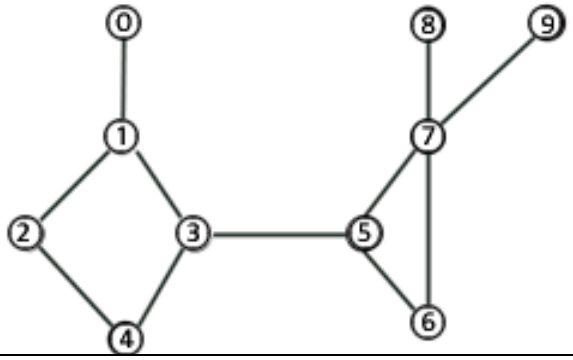
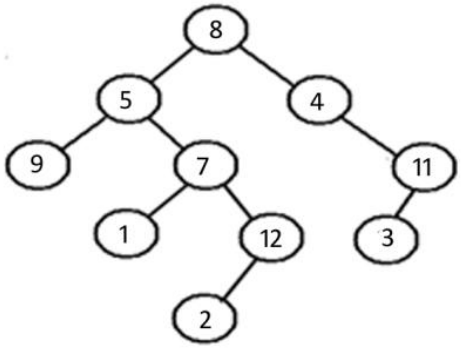
| | | |
|---|---|---|
| **Course Name** | : | **DESIGN AND ANALYSIS OF ALGORITHMS** |
| **Course Code** | : | A40508 |
| **Class** | : | II B. Tech II Semester |
| **Branch** | : | Computer Science and Engineering |
| **Year** | : | 2016 – 2017 |
| **Course Faculty** | : | Dr. L V Narasimha Prasad , Mr. Y Subba Rayudu Assistant Professor<br>Mrs. G.Vasavi Assistant Professor |

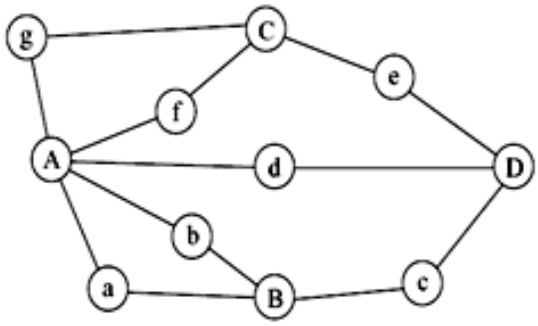
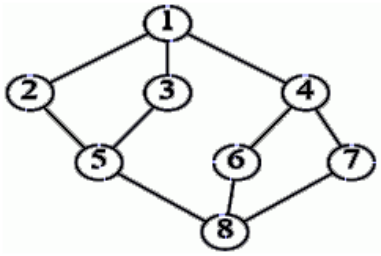
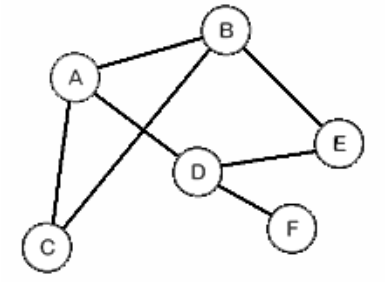
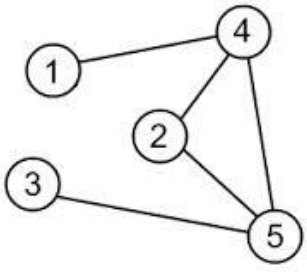
**OBJECTIVES**

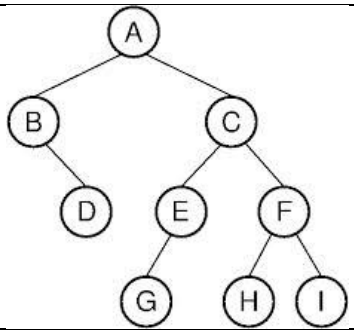
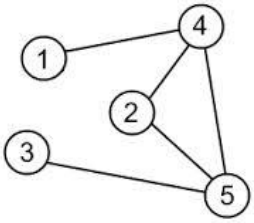
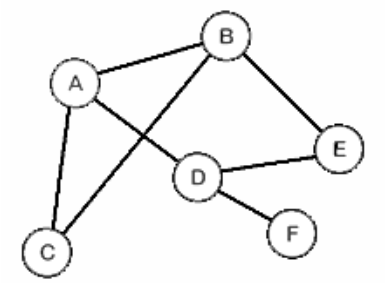
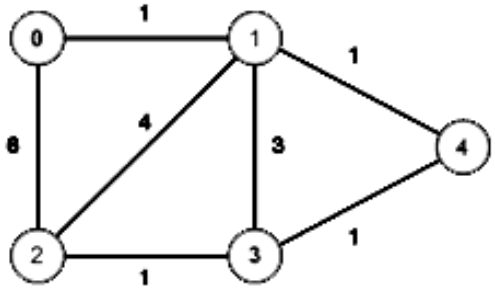
To meet the challenge of ensuring excellence in engineering education, the issue of quality needs to be addressed, debated and taken forward in a systematic manner. Accreditation is the principal means of quality assurance in higher education. The major emphasis of accreditation process is to measure the outcomes of the program that is being accredited.

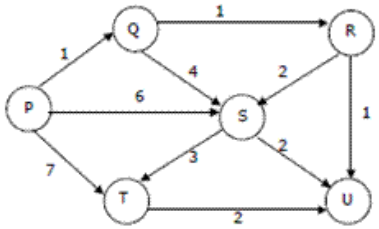
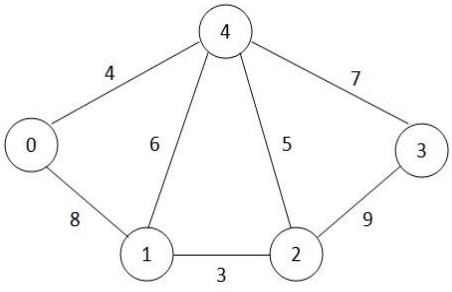
In line with this, Faculty of Institute of Aeronautical Engineering, Hyderabad has taken a lead in incorporating philosophy of outcome based education in the process of problem solving and career development. So, all students of the institute should understand the depth and approach of course to be taught through this question bank, which will enhance learner's learning process.

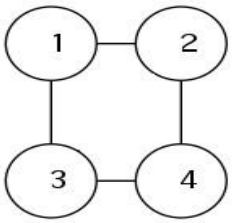
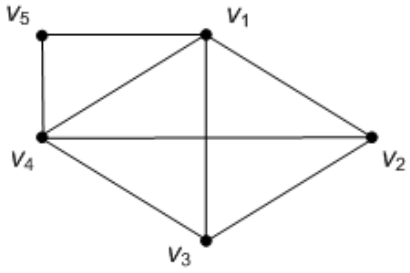
| S. NO. | QUESTION | BLOOMS TAXONOMY LEVEL | PROGRAM OUTCOME |
|---|---|---|---|
| | **UNIT – I** | | |
| 1 | **Solve** the following recurrence relation $T(n) = \left\{2T\left(\frac{n}{2}\right) + n, \quad \text{and} T(1) = 2\right.$ | Apply | 4 |
| 2 | **Solve** the following recurrence relation $T(n) = 7T(n/2)+cn^2$ | Apply | 4 |
| 3 | **Solve** the recurrence relation $T(n) = \begin{cases} k, & n = 1 \\ 3T\left(\frac{n}{2}\right) + kn, & n > 1, \quad n \text{ is powerof } 2 \end{cases}$ | Apply | 4 |
| 4 | **Explain** quick sort algorithm and trace the algorithm for following data sequence:<br>3, 5, 9, 7, 1, 4, 6, 8, 2 | Apply | 7 |
| 5 | **Sort** the list of numbers using merge sort<br>33, 44, 2, 10, 25, 79, 86, 47, 14, 36 | Apply | 7 |

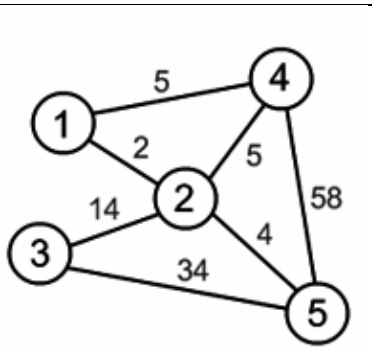
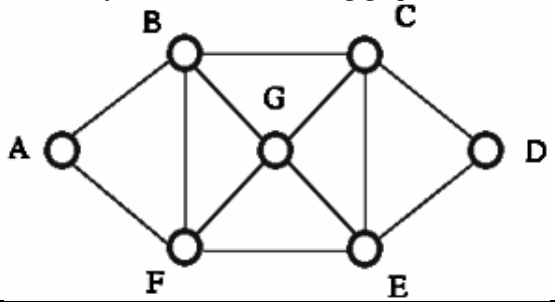
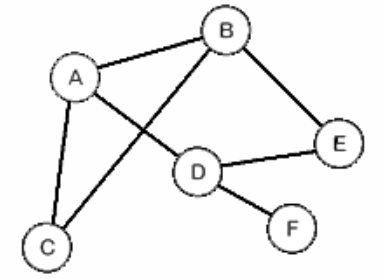
| S. NO. | QUESTION | BLOOMS TAXONOMY LEVEL | PROGRAM OUTCOME |
|---|---|---|---|
| 6 | **Show** that the average case time complexity of quick sort is O(nlogn) | Apply | 7 |
| 7 | **Understand** merge sort on letters H, K, P,C,S,K,R,A,B,L | Apply | 7 |
| 8 | **Understand**strassen's matrix multiplication on following matrices $\begin{bmatrix} 4 & 5 \\ 5 & 9 \end{bmatrix}, \begin{bmatrix} 2 & 10 \\ 1 & 6 \end{bmatrix}$ | Apply | 7 |
| 9 | **Write** and solve recurrence relation for strassen's matrix multiplication | Apply | 7 |
| 10 | **Solve** the following recurrence relation $T(n) = \begin{cases} 2T\left(\frac{n}{2}\right) + 1, & \text{and} T(1) = 2 \end{cases}$ | Apply | 4 |
| | **UNIT – II** | | |
| 1 | **Illustrate** BFS traversal of following graph  | Apply | 5 |
| 2 | **List** the articulation points from the following graph  | Understand | 5 |
| 3 | **Write**inorder, preoreder, post order traversal of the following tree  | Understand | 5 |

| S. NO. | QUESTION | BLOOMS TAXONOMY LEVEL | PROGRAM OUTCOME |
|---|---|---|---|
| 4 | **Illustrate** DFS and BFS traversals of following graph   | Apply | 5 |
| 5 | **Illustrate** DFS traversal of following graph   | Apply | 5 |
| 6 | **Illustrate** BFS traversal of following graph   | Understand | 5 |
| 7 | **List** the articulation points from the following graph   | Understand | 5 |
| 8 | **Write**inorder, preoreder, post order traversal of the following tree | Understand | 5 |

| S. NO. | QUESTION | BLOOMS TAXONOMY LEVEL | PROGRAM OUTCOME |
|---|---|---|---|
| |  | | |
| 9 | **Illustrate** BFS and DFS traversals of following graph  | Understand | 5 |
| 10 | **Illustrate** DFS traversal of following graph  | Understand | 5 |
| | **UNIT – III** | | |
| 1 | **Compute** the optimal solution for Job Sequencing with Deadlines using greedy method for N=4, profits: (p1,p2,p3,p4) = (100,10,15,27) Deadlines: (d1,d2,d3,d4) = (2,1,2,1) | Apply | 8 |
| 2 | **Compute** the optimal solution for Knapsack problem using greedy method for N=3, M=20, (p1,p2,p3)= (25,24,15), (w1,w2,w3) = (18,15,10) | Apply | 8 |
| 3 | **Construct** minimum cost spanning tree using a) Prims algorithm b) Kruskal algorithm  | Apply | 8 |

| S. NO. | QUESTION | BLOOMS TAXONOMY LEVEL | PROGRAM OUTCOME |
|---|---|---|---|
| 4 | **Understand** single source shortest path algorithm for the following graph   | Apply | 8 |
| 5 | **Use** optimal binary search tree algorithm and compute 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. | Apply | 8 |
| 6 | **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) | Apply | 8 |
| 7 | **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. | Apply | 8 |
| 8 | **Solve** the solution for 0/1 knapsack problem using dynamic programming N=3, m=6 profits: (p1,p2,p3) = (1,2,5) and weights: (w1,w2,w3) = (2,3,4) | Apply | 8 |
| 9 | **Find** the shortest tour of traveling sales person for the following cost matrix using dynamic Programming $\begin{bmatrix} \infty & 12 & 5 & 7 \\ 11 & \infty & 13 & 6 \\ 4 & 9 & \infty & 18 \\ 10 & 3 & 2 & \infty \end{bmatrix}$ | Apply | 8 |
| 10 | **Calculate** shortest distances using all pairs shortest path algorithm   | Apply | 9 |
| | **UNIT – IV** | | |
| 1 | **Sketch** the state space tree degenerated by 4 queens problem | Knowledge | 10 |
| 2 | **Understand** the backtracking algorithm to solve the following instance of the sum of subsets problem S={5,10,12,13,15,18} and d=30 | Apply | 10 |

| S. NO. | QUESTION | BLOOMS TAXONOMY LEVEL | PROGRAM OUTCOME |
|---|---|---|---|
| 3 | **Sketch** the state space tree generated all possible 3-color,4-node graph   | | 10 |
| 4 | **Identify** Hamiltonian cycle from the following graph   | Knowledge | 10 |
| 5 | **Solve** the following instance of travelling sales person problem using Least Cost Branch Bound $\begin{bmatrix} \infty & 12 & 5 & 7 \\ 11 & \infty & 13 & 6 \\ 4 & 9 & \infty & 18 \\ 10 & 3 & 2 & \infty \end{bmatrix}$ | Apply | 10 |
| 6 | **Draw** the portion of state space tree generated by LCBB by the following knapsack problem for 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 | 11 |
| 7 | **Draw** the portion of state space tree generated by FIFO knapsack instance : N=4, (P1, P2, P3, P4)= ( 10, 10, 12, 18 ), (W1, W2, W3,W4) = ( 2, 4, 6, 9 ) , M=15 | Understand | 11 |
| 8 | **Solve** the following instance of travelling sales person problem using Least Cost Branch Bound | Understand | 11 |

| S. NO. | QUESTION | BLOOMS TAXONOMY LEVEL | PROGRAM OUTCOME |
|---|---|---|---|
| |  | | |
| 9 | **Identify** Hamiltonian cycle from the following graph  | Understand | 10 |
| 10 | **Understand** the backtracking algorithm to color the following graph  | Understand | 10 |
| | **UNIT – V** | | |
| 1 | **State** and prove cook's theorem | Knowledge | 12 |
| 2 | **Explain** deterministic and non-deterministic algorithms | Knowledge | 12 |
| 3 | **Write** non deterministic algorithm for sorting and searching | Knowledge | 12 |
| 4 | **Write** a non-deterministic knapsack algorithm | Knowledge | 12 |
| 5 | **Explain** P and NP problems are related | Understand | 12 |
| 6 | **Distinguish** NP-hard and NP-complete problems | Knowledge | 12 |
| 7 | **Explain** decision problem with an example | Understand | 12 |
| 8 | **Explain** chromatic number decision problem and clique decision problem | Understand | 12 |
| 9 | **Explain** the strategy to prove that a problem is NP-hard | Understand | 12 |
| 10 | **Explain** intractable problems with examples | Understand | 12 |

**HOD, COMPUTER SCIENCE AND ENGINEERING**