



# INSTITUTE OF AERONAUTICAL ENGINEERING

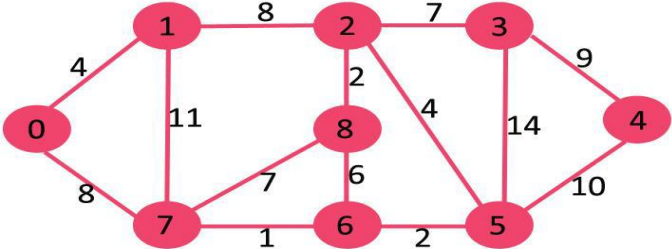
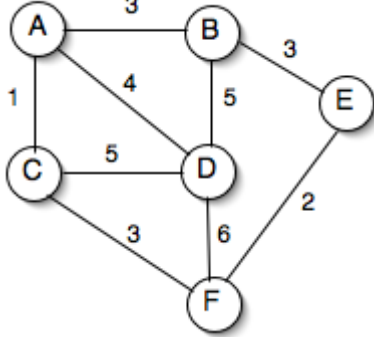
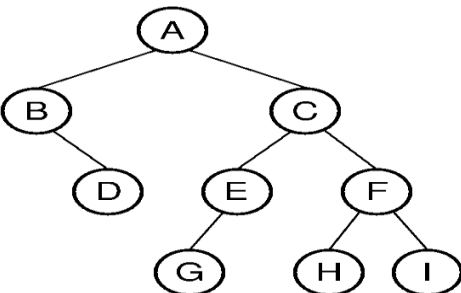
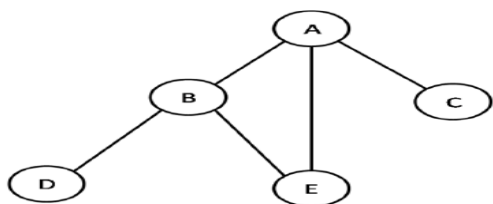
(Autonomous)

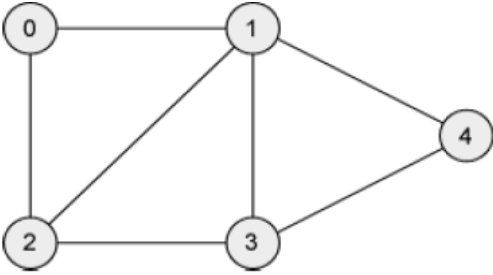
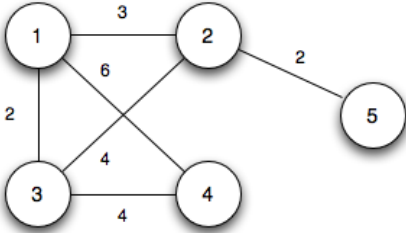
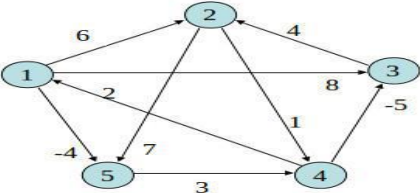
Dundigal, Hyderabad – 500043

## Information Technology

### List of Laboratory Experiments

DESIGN AND ANALYSIS OF ALGORITHMS LABORATORY								
Course Code	Category	Hours / Week			Credits	Maximum Marks		
ACSC15	Core	L	T	P	C	CIA	SEE	Total
		0	0	3	1.5	30	70	100
Contact Classes: Nil	Tutorial Classes: Nil	Practical Classes: 36			Total Classes:36			
Branch: IT	Semester: IV	Academic Year: 2021-22			Regulation: UG20			
<p><b>Course overview:</b> Design and analysis of algorithms is the process of finding the computational complexity of algorithms. It helps to design and analyze the logic on how the algorithm will work before developing the actual code for a program. It focuses on introduction to algorithm, asymptotic complexity, sorting and searching using divide and conquer, greedy method, dynamic programming, backtracking, branch and bound. NP-hard and NP-complete problems. The applications of algorithm design are used for information storage, retrieval, transportation through networks, and presentation to users.</p>								
<p><b>Course objectives:</b>  <b>The students will try to learn:</b></p> <p>I. The problem analysis and design the solution for the given problem.            II. The suitable algorithm for the given real world problem.</p>								
<p><b>Course outcomes:</b></p> <p><b>CO 1:</b> Apply Divide and conquer strategy to organize the data in ascending or descending order.  <b>CO 2:</b> Make use of Algorithmic Design paradigms to determine shortest distance and transitive closure of Directed or Undirected Graphs and Trees.  <b>CO 3:</b> Utilize Greedy Technique for generating minimum cost spanning tree of a Graph.  <b>CO 4:</b> Analyze Principle of optimality in finding solutions to optimization problems.  <b>CO 5:</b> Compare the efficiencies of traversal problems using different Tree and Graph traversal algorithms.  <b>CO 6:</b> Examine Branch and Bound Approach for solving Combinatorial optimization problems.</p>								
WEEK NO	EXPERIMENT NAME							CO
WEEK – I	<b>QUICK SORT</b>							CO 1
	Sort a given set of elements using the quick sort method and determine the time required to sort the elements. Repeat the experiment for different values of n, the number of elements in the 1st to be sorted and plot a graph of the time taken versus n. The elements can be read from a file or can be generated using the random number generator.							
WEEK – II	<b>MERGE SORT</b>							CO 1
	Implement merge sort algorithm to sort a given set of elements and determine the time required to sort the elements. Repeat the experiment for different values of n, the number of elements in the list to be sorted and plot a graph of the time taken versus n. The elements can be read from a file or can be generated using the random number generator.							

<p><b>WEEK – III</b></p>	<p><b>KNAPSACK PROBLEM</b></p>	<p><b>CO 1</b></p>
<p>Implement 0/1 Knapsack problem using Dynamic Programming.</p>		
<p><b>WEEK – IV</b></p>	<p><b>SHORTEST PATHS ALGORITHM</b></p> <p>From a given vertex in a weighted connected graph, find shortest paths from 0 to other vertices using Dijkstra's algorithm.</p> 	<p><b>CO 1</b></p>
<p><b>WEEK – V</b></p>	<p><b>MINIMUM COST SPANNING TREE</b></p> <p>Find Minimum Cost Spanning Tree of a given undirected graph using Kruskal's algorithm.</p> 	<p><b>CO 2</b></p>
<p><b>WEEK – VI</b></p>	<p><b>TREE TRAVERSALS</b></p> <p>Perform various tree traversal algorithms for a given tree.</p> 	<p><b>CO 2</b></p>
<p><b>WEEK – VII</b></p>	<p><b>GRAPH TRAVERSALS</b></p> <p>a. Print all the nodes reachable from a given starting node in a digraph using BFS method.</p> 	<p><b>CO 2</b></p>

	<p>b. Check whether a given graph is connected or not using DFS method</p> 																																					
<b>WEEK -VIII</b>	<b>SUM OF SUB SETS PROBLEM</b>																																					
	Find a subset of a given set $S = \{s_1, s_2, \dots, s_n\}$ of $n$ positive integers whose sum is equal to a given positive integer $d$ . For example, if $S = \{1, 2, 5, 6, 8\}$ and $d = 9$ there are two solutions $\{1, 2, 6\}$ and $\{1, 8\}$ . A suitable message is to be displayed if the given problem instance doesn't have a solution.	<b>CO 3</b>																																				
<b>WEEK - IX</b>	<b>TRAVELLING SALES PERSON PROBLEM</b>																																					
	Implement any scheme to find the optimal solution for the Traveling Sales Person problem and then solve the same problem instance using any approximation algorithm and determine the error in the approximation.	<b>CO 4</b>																																				
<b>WEEK - X</b>	<b>MINIMUM COST SPANNING TREE</b>																																					
	Find Minimum Cost Spanning Tree of a given undirected graph using Prim's algorithm.																																					
		<b>CO 4</b>																																				
<b>WEEK - XI</b>	<b>ALL PAIRS SHORTEST PATHS</b>																																					
	Implement All-Pairs Shortest Paths Problem using Floyd's algorithm.																																					
	 <table border="1" data-bbox="853 1512 1262 1720"> <thead> <tr> <th></th> <th>1</th> <th>2</th> <th>3</th> <th>4</th> <th>5</th> </tr> </thead> <tbody> <tr> <th>1</th> <td>0</td> <td>6</td> <td>8</td> <td><math>\infty</math></td> <td>-4</td> </tr> <tr> <th>2</th> <td><math>\infty</math></td> <td>0</td> <td><math>\infty</math></td> <td>1</td> <td>7</td> </tr> <tr> <th>3</th> <td><math>\infty</math></td> <td>4</td> <td>0</td> <td><math>\infty</math></td> <td><math>\infty</math></td> </tr> <tr> <th>4</th> <td>2</td> <td><math>\infty</math></td> <td>-5</td> <td>0</td> <td><math>\infty</math></td> </tr> <tr> <th>5</th> <td><math>\infty</math></td> <td><math>\infty</math></td> <td><math>\infty</math></td> <td>3</td> <td>0</td> </tr> </tbody> </table>		1	2	3	4	5	1	0	6	8	$\infty$	-4	2	$\infty$	0	$\infty$	1	7	3	$\infty$	4	0	$\infty$	$\infty$	4	2	$\infty$	-5	0	$\infty$	5	$\infty$	$\infty$	$\infty$	3	0	<b>CO 5</b>
	1	2	3	4	5																																	
1	0	6	8	$\infty$	-4																																	
2	$\infty$	0	$\infty$	1	7																																	
3	$\infty$	4	0	$\infty$	$\infty$																																	
4	2	$\infty$	-5	0	$\infty$																																	
5	$\infty$	$\infty$	$\infty$	3	0																																	
<b>WEEK - XII</b>	<b>N QUEENS PROBLEM</b>																																					
	Implement N Queen's problem using Back Tracking	<b>CO 6</b>																																				