Name of the Faculty: Ms. Neha
Discipline: Computer science & Engg.
Semester:5th
Subject: Design Algorithm & Analysis(CSE 305N)
Work Load (Lecture/Practical) per week (in hours): Lectures- 3 Hours , Practicals- 3 Hours

	Theory		Practical	
Week	**Lecture day**	**Topic (including assignment/test)**	**Practical day**	**Topic**
1st	**1st**	**Review**: Elementary Data Structures	**1st**	Sort a given set of elements using the Quick sort method and determine the time required to sort the elements. The elements can be read from a file or can be generated using the random number generator.
	2nd	**Review**:Algorithms & its complexity(Time & Space		
	3rd	**Review**:Analysing Algorithms, Asymptotic Notations,		
2nd	**4th**	**Review:**Priority Queue, Quick Sort	**2nd**	Program to Obtain the Topological ordering of vertices in a given digraph.
	5th	**Review**:merge sort.(Assignment-1)		
	6th	**Recurrence relation**:Methods for solving recurrence		
3rd	**7th**	**Recurrence relation:**Substitution , Recursion tree	**3rd**	Implement 0/1 Knapsack problem using Dynamic Programming.
	8th	**Recurrence relation:** Master theorem, Strassen multiplication.		
	9th	**Advanced data Structures:** Binomial heaps		
4th	**10th**	**Advanced data Structures:**Fibonacci heaps, Splay Trees	**4th**	From a given vertex in a weighted connected graph, find shortest paths to other vertices using Dijkstra's algorithm.
	11th	**Advanced data Structures:**Red-Black Trees		
	12th	**Unit test-I**		
5th	**13th**	**Advanced Design and analysis Techniques**	**5th**	Find Minimum Cost Spanning Tree of a given undirected graph using Kruskal's algorithm.
	14th	**Dynamic programming**:: Elements, Matrix-chain multiplication		
	15th	**Dynamic programming**:longest common subsequence,		
6th	**16st**	**Greedy algorithms:** Elements	**6th**	Program to Print all the nodes reachable from a given starting node in a digraph using BFS method.
	17nd	**Greedy algorithms:** Activity- Selection problem,		
	18rd	**Greedy algorithms:** Huffman codes		
7th	**19th**	**Greedy algorithms:**Task scheduling problem, Travelling Salesman Problem. (Assignment-2)	**7th**	Compute the transitive closure of a given directed graph using Warshall's algorithm.
	20th	**Backtracking algorithms:** Graph coloring		
	21th	**Backtracking algorithms:** N-Queen problem,		
8th	**22th**	**Backtracking algorithms:** Hamiltonian path and circuit	**8th**	Program to Check whether a given graph is connected or not using DFS method.
	23th	**Unit test-II**		
	24st	Graph Algorithms :Introduction		
9th	**25rd**	**Review of graph algorithms**	**9th**	Find Minimum Cost Spanning Tree of a given undirected graph using Prim's algorithm.
	26th	**Review of graph algorithms:**Traversal Methods(Depth first & Breadth first search),		
	27th	**Review of graph algorithms:**Topological sort,		
10th	**28th**	**Review of graph algorithms:**Strongly connected components	**10th**	Implement All-Pairs Shortest Paths Problem using Floyd's algorithm. Parallelize this algorithm, implement it using Open and determine the speed-up achieved.
	29th	**Review of graph algorithms:,** Minimum spanning trees- Kruskal's		
	30th	**Review of graph algorithms:**Prim's Algorithm		
11th	**31st**	**Review of graph algorithms:**Single source shortest paths, Relaxation	**11th**	Implement N Queen's problem using Back Tracking
	32nd	**Review of graph algorithms:**Dijkstra's Algorithm(Assignment-3)		
	33rd	**Review of graph algorithms**:Bellman- Ford algorithm(Assignment-3)		
12th	**34th**	**Review of graph algorithms:**Single source shortest paths for directed acyclic graphs,	**12th**	Implement Graph Coloring.
	35th	**Review of graph algorithms:**Floyd-Warshall algorithm		
	36th	**Unit Test -III**		
	37th	**Computational Complexity**:Basic Concepts		Find Hamiltonian Path using Back Tracking.

13th	**38th**	**Computational Complexity**:Polynomial vs Non-Polynomial Complexity	**13th**	
	39st	**Computational Complexity**:NP- hard & NP-complete classes.		
14th	**40rd**	**Computational Complexity**:Flow and Sorting Networks,Ford- Fulkerson method	**14th**	Implement longest common subsequence.
	41`th	**Computational Complexity**:Ford- Fulkerson method(Assignment-4)		
	42th	**Computational Complexity**:Comparison network		
15th	**43th**	**Computational Complexity**:Zero- one principle	**15th**	Implement Huffman code using Greedy approach
	44th	**Computational Complexity**:Bitonic sorting network, merging network		
	45th	**Unit Test-IV**		