INDEX

Serial	Assignments	Page no.
1.	Implementation of graph using adjacency matrix. a) Undirected Unweighted b) Undirected Weighted c) Directed Unweighted d) Directed Weighted	1 - 6
2.	Implementation of graph using incidence matrix. a) Undirected Unweighted b) Undirected Weighted c) Directed Unweighted d) Directed Weighted	7 - 10
3.	Implementation of graph using adjacency list. a) Undirected Unweighted b) Undirected Weighted c) Directed Unweighted d) Directed Weighted	11 - 18
4.	a) Consider an undirected or directed graph, do DFS on it, and compute the DFS tree. b) Compute the no of components of a given undirected graph using DFS. c) Consider an undirected or directed graph, do DFS on it, & differentiate its edges based on your sequence visiting the vertices.	19 - 33
5.	a) Consider an undirected or directed graph, do BFS on it, and compute the BFS tree. b) Do BFS for computing the shortest distance and the path between a pair of vertices of an undirected or directed graph, if one exists.	34 - 38
6.	Devise a static scheme in computing a polynomial ‘C’ where ‘C’ is computed by a) Adding two polynomials ‘A’ & ‘B’. b) Subtracting polynomial ‘B’ from ‘A’. c) Multiplying two polynomial ‘A’ & ‘B’. d) Differentiating polynomial ‘A’.	39 - 46
7.	Devise a scheme a sparse matrix to represent a sparse matrix M and transpose this representation of M in lexicographic order.	48 – 51
8.	Devise a dynamic scheme in computing a polynomial ‘C’ where ‘C’ is computed by a) Adding two polynomials ‘A’ & ‘B’. b) Subtracting polynomial ‘B’ from ‘A’. c) Multiplying two polynomial ‘A’ & ‘B’. d) Differentiating polynomial ‘A’.	36 - 41
9.	Consider a sequence of n elements and construct an AVL tree after insertion of each element into the tree. Also perform deletion of elements from the AVL tree and rebuild the AVL tree as needed.	52 - 63

10.	Consider a sequence of n elements and construct a B-tree after insertion of each element into the tree. Also perform deletion of elements from B-tree and obtain the resulting B-tree after each deletion.	64 – 70
11.	Implement Heap Sort	71 – 73
12.	Implement Shell Sort	74 – 76
13.	Implement Graph Sort	77 – 80
14.	Implement Sieve Sort	81 – 83
15.	Consider the problem of implementing n digit binary integers and compute amortized cost of this problem in terms of n.	84 – 87
16.	Perform the operation of ‘Binomial_Heap_Extract_Min’ for a Binomial Heap that comprises almost 6 binomial trees where minimum key value pressed in a node that belonging to the target tree as next to the largest tree.	88 – 92