QPC: 23RAPhD015 AY 23 Reg. No

GIET UNIVERSITY, GUNUPUR – 765022

Ph.D. (Second Semester) Examinations, April – 2024

PPECS2022 - Advance Algorithms

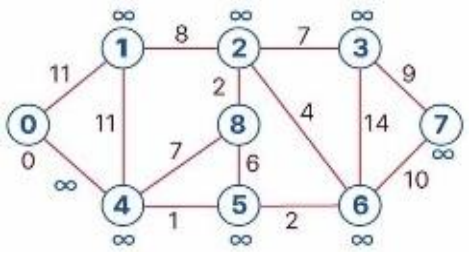
(CSE)

Time: 3 hrs Maximum: 70 Marks

The figures in the right-hand margin indicate marks.

Answer ANY FIVE Questions **(14 x 5 = 70 Marks)**

		Marks
1.a.	Show that the running time of QUICKSORT is (n lg n) when all elements of array A have the same value.	7
b.	Illustrate the Merge sort algorithm and discuss its time complexity.	7
2.a.	How Breadth First Search differ from Depth First search? Write the algorithm for Breadth First traversal of a graph.	7
b.	Apply Dijkstra's Shortest path algorithm in following graph. Take 0 as the source vertex. Illustrate all your work stepwise.	7



3.a.	Explain how the greedy paradigm is applied to solve the MWIS problem.	7
b.	Discuss the divide-and-conquer strategy employed in Strassen's algorithm, focusing on how it divides the matrices into submatrices and combines the results to obtain the final product. Provide a step-by-step illustration of this process for multiplying 2 x 2 matrices.	7
4.a.	Describe the Ford-Fulkerson algorithm for computing maximum matching.	7
b.	Explain Edmonds' Blossom algorithm for computing augmenting paths in a graph. Discuss the key steps involved in the algorithm, its time complexity analysis, and the significance of blossom contraction. Additionally, illustrate the algorithm with a suitable example and discuss its correctness.	7
5.a.	Explain the Maxflow-mincut theorem in the context of network flow theory. Describe the theorem statement, including the relationship between the maximum flow in a network and the minimum cut. Provide a proof outline for the theorem, discussing the key steps involved.	7
b.	Discuss Strassen's algorithm as an example of the divide and conquer paradigm in algorithm design. Explain the key concepts of divide and conquer, and how they are applied in Strassen's algorithm for matrix multiplication.	7
6.a.	Explain the Floyd-Warshall algorithm for finding the shortest paths in a weighted graph. Describe the algorithm's approach, including its initialization, iterative updates, and termination conditions. Analyze the time complexity of the algorithm and discuss its suitability by taking an example.	7
b.	Explain the concept of dynamic programming in algorithm design. Discuss the key characteristics of dynamic programming problems and explain how they differ from other problem-solving techniques. Provide examples of problems that can be solved using dynamic programming and explain the general approach to solving them.	7

7.a.	Discuss the concepts of NP-hardness and NP-completeness in the context of computational complexity theory. Explain the difference between NP-hard and NP-complete problems, and provide examples of each.	7
b.	Discuss the concept of approximation algorithms in the context of optimization problems. Provide examples of optimization problems where approximation algorithms are commonly applied, and describe the general approaches used to design such algorithms. Additionally, analyze the approximation ratio and time complexity considerations for approximation algorithms.	7
8.a.	Provide examples of computational problems where number theoretic algorithms play a crucial role, such as integer factorization, modular arithmetic, and primality testing.	7
b.	Provide examples of computational problems where number theoretic algorithms are indispensable, such as integer factorization, modular arithmetic, and primality testing.	7

---End of Paper---