PVP 14

Code: CS4T2

II B.Tech - II Semester – Regular/Supplementary Examinations
October-2020

DESIGN AND ANALYSIS OF ALGORITHMS
(COMPUTER SCIENCE & ENGINEERING)

Duration: 3 hours Max. Marks: 70

PART – A

Answer ***all*** the questions. All questions carry equal marks

11 x 2 = 22 M

1.

a) What are the fundamental steps involved in algorithmic problem solving?
b) What are the steps involved in the analysis framework?
c) Give the general plan for divide-and-conquer algorithms.
d) Define Recurrence Relation.
e) Discuss feasible solution, optimal solution and objective functions with example.
f) Distinguish between Prim's and kruskal's algorithm.
g) Define Principle of optimality.
h) How to solve a Dynamic Programming Problem?
i) What are the two types of constraints used in backtracking?
j) What is Hamiltonian cycle in an undirected graph?
k) Define NP, NP hard and NP complete. Give example of each.

PART – B

Answer any ***THREE*** questions. All questions carry equal marks.
3 x 16 = 48 M

2. a) Write & explain with example for asymptotic notations used for best case, average case and worst case analysis of algorithms. 9 M

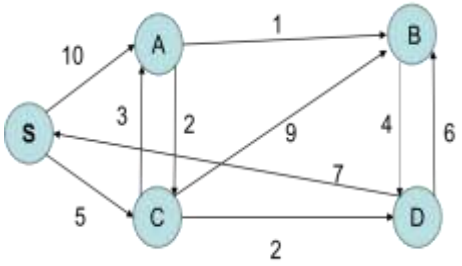
b) Write an algorithm for finding maximum element in an array. Give best, worst and average case complexities. 7 M

3. a) Perform binary search on list of elements to find the key element using divide and conquer, and also estimate the time complexity. 8 M

b) Show that the average case time complexity of quick sort algorithm is O(n logn). 8 M

4. a) Discuss general characteristics of greedy method. Mention any two examples of greedy method that we are using in real life. 6 M

b) Consider the directed edge-weighted graph shown below



Show the execution of Dijkstra's shortest path algorithm (pseudocode given below) for solving the Single Source Shortest Path (SSSP) problem on this graph. Use the vertex S as the source. 10 M

5. a) Write and explain an algorithm to compute the all pairs shortest path using dynamic programming and prove that it is optimal with an example. 8 M

b) Solve the following instance of 0/1 KNAPSACK problem using Dynamic programming.
n = 3, (W1,W2,W3) = (2,3,4), (P1,P2,P3) = (1,2,5), and m = 6. 8 M

6. a) Write the algorithm for general iterative backtracking method and explain various factors that define the efficiency of backtracking. 8 M

b) Give the formulation of modified knapsack problem using branch and bound and find the optimal solution using Least Cost Branch and Bound (LCBB) with n=4, m=15, (p1…p4) = (15,15,17,23), (w1…w4) = (3,5,6,9). 8 M