PARUL UNIVERSITY
FACULTY OF ENGINEERING & TECHNOLOGY
B.Tech. Summer 2021 - 22 Examination

Semester: 5   Date: 18/04/2022
Subject Code: 203105301   Time: 10:30am to 1:00pm
Subject Name: Design and Analysis of Algorithm   Total Marks: 60

Instructions:
1. All questions are compulsory.
2. Figures to the right indicate full marks.
3. Make suitable assumptions wherever necessary.
4. Start new question on new page.

Q1. Objective Type Questions (All are compulsory):   **15**

1. The time complexity of the following recurrence relation using Master theorem is_____.
$$T(n) = 4 \, T(n/2) + n!$$

2. Dijkstra's Algorithm cannot be applied on graph having_____. (Best/ Average/ Worst)

3. The performance of exponential time complexity is_____.

4. The asymptotic notation _____ is expressed as both the lower bound and the upper boundof an algorithm.

5. Problems that can be solved in polynomial time are known as_____?

6. $f(n) = \Theta(g(n))$ implies that?
   a. $f(n) >= c.(g(n))$ and $f(n) <= c.(g(n))$ b. $f(n) >= c.(g(n))$
   c. $f(n) <= c.(g(n))$ d. None of the above

7. Statement (I): Full bin packing provides optimal solution.
   Statement (II): Breadth First Search can be used to verify a graph contains a cycle or not.
   Which of the following is true?
   a. Both S1 and S2 are correct   b. S1 is correct and S2 is incorrect
   b. S1 is incorrect and S2 is correct   c. Both S1 and S2 are incorrect

8. Master's theorem can solve all types of recurrence.
   a. True   b. False
   c. Can't say   d. None of the above

9. Which of the given options provides the increasing order of asymptotic complexity of functions
   $2^n, n^{\frac{3}{2}}, nlogn, n^{logn}$?
   a. $2^n, n^{\frac{3}{2}}, nlogn, n^{logn}$ b. $2^n, n^{logn}, n^{\frac{3}{2}}, nlogn$
   c. $nlogn, n^{\frac{3}{2}}, n^{logn}, 2^n$ d. $n^{logn}, nlogn, n^{\frac{3}{2}}, 2^n$

10. Which of the following is true regarding Depth First Search?
    a. It is used to verify a given graph is connected or not.
    b. It can be used to find single source shortest path in the given unweighted graph.
    c. Both a and b are true
    d. None of the above

11. List two properties or characteristics of Greedy algorithm.

12. In Huffman coding, data(value) in a tree always occur at?

13. Define Backtracking.

14. Define Randomized algorithm.
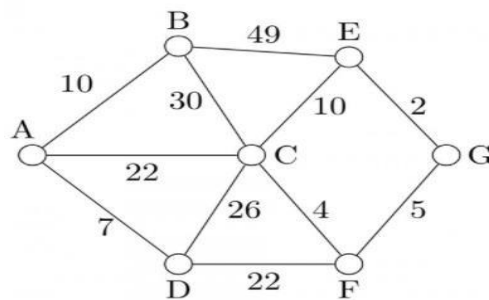
15. Define Algorithm.

Q.2 Attempt any three: **15**

(A) (i) Solve the following recurrence relation using Substitution method.

$$T(n) = 8T(n/2) + n^2 \quad \text{if } n > 2$$
$$1 \qquad\qquad\qquad \text{if } n = 2$$

(ii) Solve the following recurrence relation using Recursive tree method.
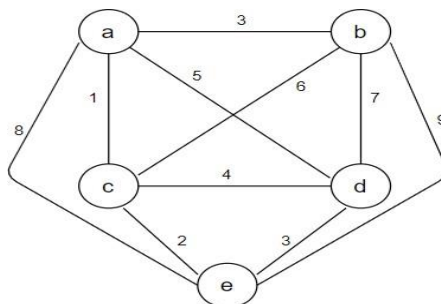
$$T(n) = T(n/2) + T(n/2) + n.$$

(B) Define MST. Find minimum spanning tree of the below diagram using Prim's algorithm.

(C) Consider the following matrix and find the minimum number of multiplications required to multiply them using Dynamic programming. Also give the optimal parenthesization of matrices.

A(1x3), B(3x5), C(5x4,) D(4x2)

(D) Consider the following diagram and find the shortest possible route that visits every node exactly once & returns to the starting point using Dynamic programming. Assume that the tour starts with vertex **'a'** and **'c'** is visited before **'b'**.

Q3. (A) Explain Backtracking method. What is N-Queens problem? Give solution of 4-Queens   **07**
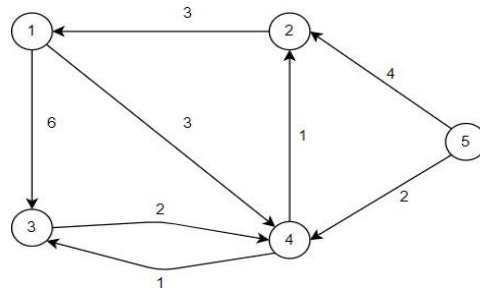problem using Backtracking method.

(B) Explain making coin change problem. Solve making coin change problem using Dynamic   **08**
programming. (Denominations: d1=1, d2=4, d3=6, d4=7). Give your answer for making
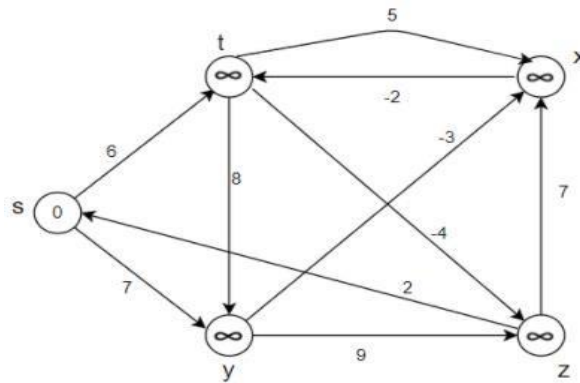change of Rs. 9.

OR

(B) Explain Floyd-warshall algorithm. Find all pairs shortest path of the following diagram by applying Floyd-warshall algorithm.   **08**

Q4. (A) Explain Bellman ford algorithm to find shortest path. Apply Bellman ford algorithm and find single source shortest path from vertex A.   **07**
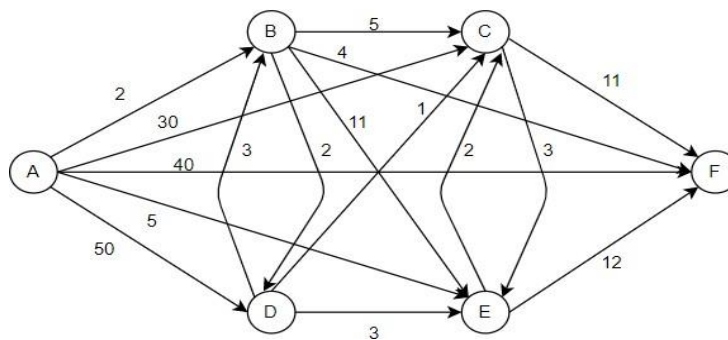

OR

(A) Explain Dijkstra algorithm. Apply Dijkstra algorithm and find single source shortest path from vertex A.   **07**

(B) Define each of the following terms. Also write the examples.   **08**
   (i) P problem
   (ii) NP problem
   (iii) NP-Hard problem
   (iv) NP Complete problem
   (v) Approximation algorithm