

Reg. No:

V/D/TY

Final Assessment Test - April 2025



VIT
Vellore Institute of Technology
Approved to the Education, under section 3 of the UVC Act, 1986

Course: BCSE204L - Design and Analysis of Algorithms
Class NBR(s): 1468/1481/1493/1499/1506/1513/1521/
1527/1531/1538/1544/1551/1558/1563/1571/1582/
1589/1597/1606

Slot: A2+TA2

Max. Marks: 100

Time: Three Hours

- KEEPING MOBILE PHONE/ANY ELECTRONIC GADGETS, EVEN IN 'OFF' POSITION IS TREATED AS EXAM MALPRACTICE
- DON'T WRITE ANYTHING ON THE QUESTION PAPER

Answer ALL Questions
(10 X 10 = 100 Marks)

1. a) Obtain an upper bound for the following recurrence equation using recursive tree method and verify using Master's theorem [6+4]

$$T(n) = \begin{cases} 1 & \text{when } n = 1 \\ 3T\left(\frac{n}{4}\right) + n^2 & \text{when } n > 1 \end{cases}$$

- b) Solve the following Knapsack problem which consists of divisible items using Greedy strategy. Knapsack capacity is 35. Find the items to be included into the knapsack bag to gain optimal profit, where the weights of items are 10, 20, 12 with corresponding profits 60, 100, 120.

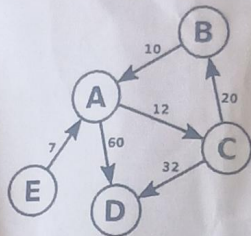
2. Design a divide and conquer algorithm to find the maximum Subarray whose sum is maximum. Consider the array of numbers given by [2, -4, 1, 9, -6, 7, -3] to illustrate the algorithm. Trace the steps of the algorithm with the help of a recursive tree. Discuss its time complexity.

3. Design a dynamic programming algorithm for paranthesization of matrix chain which will require minimum number of scalar multiplications to find the product. Consider the chain of matrices given by A(2x3) B(3x2) C(2x4) and D(4x3) to illustrate the algorithm.

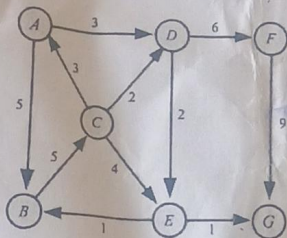
4. Solve the following instance of Knapsack problem using fixed length FIFO branch and bound strategy where we have 4 items and the knapsack bag has a capacity of 15. The profit values are {10, 12, 20, 18} and the corresponding weights are {2, 4, 5, 9}. Draw the annotated state space tree with cost and bound values. The steps for calculating bound values of various nodes should also be listed. Find the optimal solution and provide the list of items to be added into the knapsack bag.

5. Apply an algorithm that constructs longest prefix suffix (LPS) table for the pattern to find where it is plagiarized in a given text. List the algorithm and trace the steps for the pattern given by 'ABABAD' and the text given by 'ABCABABABAD'. Also return the position where the pattern is found in the text. Analyze its time complexity.

6. Find all pairs shortest path for the following weighted digraph using Floyd-Warshall's Algorithm. Analyse its time complexity. Illustrate the tracing of the algorithm.



7. Apply Ford Fulkerson's algorithm on the following network flow graph to find the maximum flow from start vertex 'A' to sink vertex 'G'. In each iteration, the graph and the residual flow network obtained after augmenting the flow should be drawn and described.



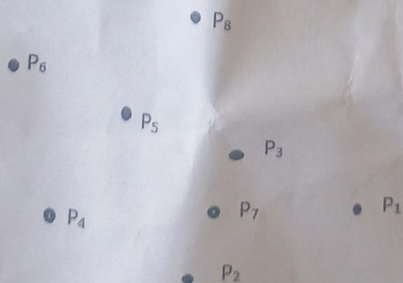
- 8.a) Design an algorithm for checking whether the two given line segments intersect or not. Trace and find if the line segment given by p_1q_1 and another line segment given by p_2q_2 intersect or not.

- $p_1(2,10)$ $q_1(7,2)$ and $p_2(4,15)$ $q_2(13,3)$
- $p_1(6,4)$ $q_1(10,16)$ and $p_2(8,10)$ $q_2(13,6)$
- $p_1(6,1)$ $q_1(7,3)$ and $p_2(8,4)$ $q_2(6,7)$
- $p_1(2,3)$ $q_1(10,3)$ and $p_2(5,2)$ $q_2(7,3)$

OR

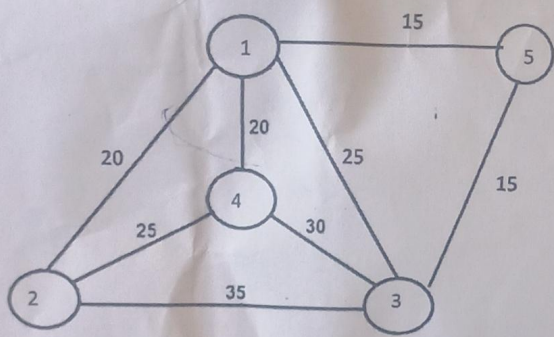
- 8.b) Consider the following set of points 'Q' in a XY plane. Construct a smallest polygon enclosing all the points. Design a $O(n \log n)$ time algorithm to find the same and trace it using the following points. Elaborate the step-by-step process of constructing the polygon.

$$Q = \{P_1(10,3), P_2(6,1), P_3(8,4), P_4(2,2), P_5(3,5), P_6(1,6), P_7(7,3), P_8(6,7)\}.$$



9 Apply Randomized quicksort algorithm by selecting the random pivot to trace the steps to sort in detail. Discuss how the average time complexity of randomized quicksort is better than the worst-case time complexity of deterministic quicksort. The elements are given below:
 99,34,46,22,60,23,15,14,33,20

- 10.(a) i. Prove that the approximation algorithm for solving Travelling Salesman problem is a polynomial time 2-approximation algorithm. [4]
 ii. Design a polynomial time approximation algorithm for the travelling salesman problem. Illustrate the algorithm for the graph given below. Let 1 be the start vertex. [6]



OR

10.(b) Prove that "3-CNF (Conjunctive Normal Form) SAT" is NP complete. Use the satisfiability formula given below to prove the NP completeness. Illustrate the three steps for conversion of ϕ to ϕ''' . For two clauses, the detailed steps for converting ϕ' to ϕ''' are desired.

$$\phi = (x_4 \wedge ((x_1 \rightarrow x_2) \wedge (\bar{x}_3 \Leftrightarrow x_1)))$$

⇔⇔⇔ V/D/TY ⇔⇔⇔