DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

NITK-SURATHKAL

Sub: Algorithms and Complexity

Surprise Quiz :01 Max Marks : 25 Duration: 45minutes

Note: Answer all the questions. Missing data may be suitably assumed.

1. Consider the problem of finding the kth smallest element from an unordered array discussed in the class. We had divided the set into subsets of size 5. Can we divide the set into subsets of 7 or 10? What is the time complexity of the algorithm when we use the subset size as 5 or 7. Derive the time complexity. ---------- 10 Marks
2. Minimum Cost Polygon Triangulation
   A convex polygon with nnn vertices can be divided into non-overlapping triangles by drawing non-intersecting diagonals. The cost of a triangulation is defined as the sum of the perimeters of the triangles formed.
   (a) [1Marks] Explain why the problem of finding the minimum cost triangulation of a convex polygon exhibits *optimal substructure* and *overlapping subproblems*.
   (b) [4 Marks] Formulate a dynamic programming solution. Clearly define the DP state, recurrence relation, and base case. State the overall time and space complexity of your algorithm.
   (c) [4 Marks] Using your recurrence, compute the minimum cost triangulation for the following quadrilateral: (0,0), (1,0), (1,1), (0,1)
   (d) [1 Marks] Compare this problem with the Matrix Chain Multiplication problem. *Highlight* one similarity and one difference.

3. You are given n sorted files of sizes f1,f2,…,fn. To merge two files of sizes x and y, the cost incurred is x+y After merging, the new file has size x+y. The goal is to determine the minimum total cost to merge all files into a single file.

   Tasks:
   (a) [1 Marks] Show, with an example, that merging files arbitrarily does not guarantee the minimum cost.

   (b) [1 Marks] Justify why the greedy choice of *always merging the two smallest files first* leads to an optimal solution.

   (c) [2Marks] Design a greedy algorithm using a suitable data structure and analyze its time complexity.

   (d) [1 Marks] Compare this problem with **Huffman coding**, and highlight one similarity and one difference.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

NITK-SURATHKAL

M.Tech (CSE-IS) Mid Semester Examinations

Max Marks: 25 Sub: Algorithms and Complexity Duration: 1.5 Hrs Time:10:30-12:00

Date: 06-10-2025 Course Code: CS 700 Reg No:----------

Note: Answer all the questions. Missing data may be suitably assumed.

1. You are given a set of n points in a 2D plane, each point represented by its coordinates (x,y). Design an algorithm based on the divide and conquer strategy to find the pair of points that are closest to each other in terms of Euclidean distance.
   (a) Explain how you would divide the set of points into subproblems.
   (b) Describe the steps to combine the results from the subproblems to ensure that the closest pair may lie across the dividing line.
   (c) Analyze the time complexity of your algorithm and compare it to the brute force approach. (5 Marks)

2. You are given k eggs and a building with n floors. The problem is to determine the minimum number of trials required in the worst case to find the highest floor from which an egg can be dropped without breaking.

Tasks:

a. Formulate the problem as a Dynamic Programming problem by:
   - Defining the subproblem.
   - Deriving the recurrence relation.

b. Write the DP algorithm to solve the problem.
c. Analyze the time complexity of your algorithm.
d. For the case of k = 2 eggs and n = 10 floors, construct the DP table and compute the minimum number of trials. (10 Marks)

3. You are given n items, each with a weight and value, and a knapsack of capacity W. You may take fractions of items.
   (a) Define the greedy choice property for this problem.
   (b) Prove that sorting by value/weight ratio yields the optimal solution. (05 Marks)

4. Solve the recurrence: $T(n)= 2T(\sqrt{n}) + \log n$ for $n> 2$ (03 Marks) $= \log n \times \log(\log n)$

5. You are given an array of size n. A majority element is an element that occurs more than n/2 times in the array.

a) Design a divide and conquer algorithm to find the majority element (if it exists).
b) Write the recurrence relation for the algorithm and solve its time complexity.

(02 Marks)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

NITK-SURATHKAL

Sub: Algorithms and Complexity

Max Marks: 25

1. You are given a set of people who owe each other money. The goal is to settle all debts using the minimum number of cheques (or transactions).In other words: Given debts between individuals, find the smallest set of payments that clears all debts.

Example: Suppose 3 people have these debts:

- A owes B ₹100
- A owes C ₹50
- B owes C ₹70

Without minimization → **3 cheques**. With minimization 2 cheques. Prove that this problem is in NPC (10 Marks)

2. Prove that Independent Set is Polytime reducible to Circuit Satisfiability. (10 Marks)
3. Prove that DLP decision problem is in NP. **Given** a prime p, a generator g, an element h, and an integer k, **decide** whether $g^k \equiv h \bmod p$? (5 Marks)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

NITK-SURATHKAL

M.Tech (CSE-IS) End Semester Examinations. Reg No:-25215032

Date: 28-11-2025 Time: 9:00-12:00 Max Marks: 100

Sub: Algorithms and Complexity Course Code: CS700

Note: Answer all the questions. Missing data may be suitably assumed.

1. Consider the problem of finding the *majority element* in an array of comparable elements. A majority element is defined as an element that appears more than n/2 times in an array of size n. If no such element exists, the algorithm should indicate this. Design a divide and conquer algorithm to solve the majority element problem. Derive the time complexity. (10 Marks)
2. The Maximum Submatrix Sum problem involves finding a submatrix within a given 2D matrix such that the sum of all elements within that submatrix is maximized. The elements can be positive, negative and zero. Give a divide and conquer algorithm and derive its time complexity. (15 Marks)
3. Give a dynamic programming algorithm to choose r things out of n. Derive its time complexity. Explain the algorithm with an example. (10 Marks)
4. **Minimum Number of Platforms (Train Scheduling):** You are given arrival and departure times of trains at a station. Using a greedy technique, find the minimum number of platforms required so that no train waits. Give the time complexity. (15 Marks)
5. Consider the problem of job scheduling with profits and deadlines discussed in the class. Prove that the decision problem of the same belongs to class NP. (10 Marks)
6. We are given a boolean formula in conjunctive normal form (CNF). The SAT problem is to determine if there exists an assignment of truth values to the variables that makes the entire formula true. Prove that SAT is polytime reducible to 3SAT. (15 Marks)
7. We are given a graph and we want to determine if it is 3-colorable. This is a decision problem. Prove that this problem is NP-Complete. (10 Marks)
8. Give an approximation algorithm for the set cover problem (15 Marks)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

NITK, Surathkal

Algorithms And Complexity (Code: CS700)

November 28, 2025

I Semester M.Tech CSE

Registration No. 252CS618

Time: 3Hrs. *End Semester Exam* Maximum Marks: 80

Note:- *Answer all the questions. Missing data may be suitably assumed, provided the assumption does not change the difficulty of the problem.*

1. In the algorithm SELECT(the k^{th} smallest in a set whose worst case linear time), the input elements are divided into groups of 5. Will the algorithm work in linear time, if they are divided into groups of 7? How about groups of 3? Analyze the running time of this algorithm? [06]

2. Let $X[1 \ldots n]$ and $Y[1 \ldots n]$ be two arrays each containing n numbers already in sorted order. Design and analyse an efficient divide and conquer $O(\log n)$-time algorithm to find the median of all $2n$ elements in arrays X and Y. [08]

3. Given n files of length $m_1, m_2, \ldots\ldots m_n$, the Optimal Tape Storage problem is to find which order is the best to store them on the tape, assuming that
   i) Each retrieval starts with the tape rewound.
   ii)Each retrieval takes time equal to the length of the preceding files in the tape plus the length of the retrieval file

   a) Describe a greedy algorithm for this problem.
   b) What is the running time of your algorithm?
   c) Prove your algorithm is correct. [10]

4. Given a sequence $k = k_1 < k_2 < \ldots\ldots < k_n$ of n sorted keys, with a search probability p_i for each key k_i . Build the Binary search tree that has the least search cost given the access probability for each key. Design an algorithm for the above given problem.

   Show the step by step method to construct an optimal BST for keys A, B, C, and D with search probabilities 0.1, 0.2, 0.4, and 0.3, respectively? [12]

5. Construct a maximum length common sub-sequence from the completed C table and the original sequences $X =< A, B, C, B, D, A, B >$ and $Y =< B, D, C, A, B, A >$ Show all the steps. [08]

6. Give a polynomial time reduction from CNF to 3-SAT problem. [07]

7. Assuming that Hamiltonian cycle problem is NP-Complete, prove that travelling salesman problem (TSP) is NP-Complete. [08]

8. Give a 2 - factor approximation algorithm with analysis for vertex cover problem. [10]

9. a. Solve the recurrence $T(n) = 2T(\sqrt{n}) + \log n$ and give tight bounds.
   b. Let A1, A2, A3, and A4 be four matrices of dimensions 30 × 1, 1 × 40, 40 × 10 and 10 × 25, respectively. The minimum number of scalar multiplications required to find the product $A1A2A3A4$ using the basic matrix multiplication method is .....
   c. Define NP-Completeness
   d. Consider the definition of an approximation algorithm. Explain the meaning of approximation ratio in the case of a maximisation problem.

[Each question 02]

10. Answer the following questions with either TRUE or FALSE. Explain your choice. (No credit if no explanation given.)

   (a) To prove the correctness of a greedy algorithm, we must prove that every optimal solution contains our greedy choice.
   (b) 3SAT cannot be solved in polynomial time, even if P = NP.
   (c) Repeatedly selecting a vertex of maximum degree, and deleting the incident edges, is a 2-approximation algorithm for Vertex Cover.

[01 marks each]