1	A	Two algorithms take **n^2 days** and **2^n seconds** respectively to solve an instance of size n. What is the smallest instance on which the former algorithm outperforms the later?
	B	Find the big Oh Notation for the following recurrence relation $T(n) = 3T(n-1) - 2\,T(n-2) + 3 \times 2^n$
	C	How long does it take to insert a new element into a heap? To return the smallest thing in a min-heap? To delete the smallest thing in a min-heap? To find the largest thing in a min-heap?

	D	Consider the following program outline where |A| represents the number of items in array A.
		1. method Review (array A) {
		if (|A| > 1) {
		2. Do something to A that takes time O(|A| log(|A|));
		3. Split A into two equal size pieces called B and C;
		4. Review(B); Review(C);
		5. Modify A using O(|A|) time; } }
		What is the recurrence that describes the time taken by this algorithm in terms of master theorem.

	E	Explain Boyer Moore Algorithm for String Comparison.
2	A	Let a 3-tree be a rooted tree defined inductively, as follows: A one-node rooted tree is a 3-tree of rank 0. If n > 0, 3-tree of rank n can be created from three 3-trees of rank n - 1 by attaching the roots of two of them as leftmost children of the third. Clearly, there are 3^K nodes in a 3-tree of rank k. So draw the 3-trees for a) Rank 0 b) Rank 1 c) rank 2 d) rank 3
	B	Write a method which takes two Stack objects as parameters and which returns a new Stack object which contains the contents of the first Stack, in the same order that they appear in the original, followed by the contents of the second stack in the same order that they appear in the original. Your method should leave the two original stacks unchanged when the method completes.
	C	Explain a) Josephus Permutation b) Huffman Code compression c) Genetic Programming
3	A	Suppose you come up with an algorithm to multiply 4 × 4 matrices with k scalar multiplications and a series of additions and subtractions. How small does k need to be for you to be able to use it to beat

		Strassen's algorithm?
	B	Using Kruskal's method of MST find the five edges

	A	B	C	D	E	F
A		7	6	1	3	12
B			4	5	8	10
C				9	11	15
D					2	14
E						13
F						

Show the edges in chronological order: Edge 1 Edge 2 Edge 3 Edge 4 Edge 5

	C	We are given an ordered sequence x 1 < x 2 < x3 < x4 <x5….< x n in linked list format. These elements have to be put into a complete binary search tree with n nodes, stored in the standard array format for complete trees. Explain how you can do this in time O(n) using just O(1) extra space
	D	What is the DFT of $A(x) = 1 + 2x^4$ when it is interpreted as a polynomial of degree- bound 8?
4	A	You are given a k×k checkerboard with a nonnegative number in each square (the square colors are not significant). A token is moved from square to square on the board. Each time the token enters a square it is charged the amount written in that square. Assume that the only legal moves are to-the-right, down, and diagonally right-down. Give an algorithm that runs in $O(k^2)$ time to find the cost of the minimum cost sequence of moves beginning in the upper left corner of the board and ending at the lower right corner.
	B	Consider the problem of searching in a sorted matrix. That is, you are given an n×n matrix A, where each entry is an integer. Each row of the matrix is sorted in ascending order, and each column is also sorted in ascending order. Given a value x, the problem is to decide whether x is stored somewhere in the array (i.e., whether there is some i and j such that A[i][j] = x). Give a divide-and-conquer algorithm for this problem. (Hint: Your algorithm needs to call itself recursively, so think carefully about the parameters required. First compare x with the element in the ``middle'' of your array) .
	C	Explain a) Parallel Programming b) Approximation algorithms

5	A	The input to this problem is a sequence *S* of integers (not necessarily positive). The problem is to find the consecutive subsequence of *S* with maximum sum using Dynamic programming . ``Consecutive" means that you are not allowed to skip numbers. For example if the input was 12, -14, 1, 23, -6, 22, -34, 13 the output would be 1, 23, -6, 22.
	B	A group of n students takes the theory & Practical sections of the AAD. Nobody is tied on either a theory score or on a practical lab score. Let's call a student a winner if nobody beat him/her on both sections of the exam. The losers are students that were beaten by somebody on both sections of the exam. Come up with an efficient algorithm for identifying the winners. Then briefly summarize the main idea so that it is to be solved considerably faster than $O(n^2)$.
6	A	You have to pick the books from A shop in Delhi, but the problem is you can't buy all the 10 books you want because you have only 2000 Rs. With you. Given the cost & need of every book , write the algorithm using backtracking.
	B	Show using tree comparison that the minimum lower bound for the Sorting/Merging technique is O(nlogn)
	C	The input to the problem consists of an ordered list of *n* words. The length of the *i*th word is w_i, that is the *i*th word takes up w_i spaces. (For simplicity assume that there are no spaces between words.) The goal is to break this ordered list of words into lines, this is called a layout. Note that you cannot reorder the words. The length of a line is the sum of the lengths of the words on that line. The ideal line length is *L*. No line may be longer than *L*, although it may be shorter. The penalty for having a line of length *K* is *L-K*. The problem is to find a layout that minimizes the total penalty. Prove or disprove that the following greedy algorithm correctly solves this problem for a and/or b. *For i= 1 to n* *Place the ith word on the current line if it fits* *else place the ith word on a new lin*e
	D	a) The total penalty is the **sum** of the line penalties. b) The total penalty is the **maximum** of the line penalties. Differentiate a) NP & NP-Complete problems b) Deterministic & Non-deterministic algorithms