**Unit I**

1. Let f(n) and g(n) be asymptotically nonnegative functions. Using the basic definition of Θ-notation, prove that max(f(n), g(n)) = Θ (f(n) + g(n))
2. Why does accessing an array element take O(1) time?
3. With neat figure give Merge sort operations on the array A={3,41,52,26,38,57,9,49} Give time analysis of Merge sort algorithm.
4. Giving example, explain the Maximum subarray problem. What does FIND-MAXIMUM-SUBARRAY return when all elements of A are negative?
5. Explain three main asymptotic notations using their mathematical representations.
6. With neat figure give Merge sort operations on the array A={3,41,82,2,38,57,9,49}
7. [What is algorithm and why analysis of it is important?](https://www.geeksforgeeks.org/what-is-algorithm-and-why-analysis-of-it-is-important/)
8. Explain three main asymptotic notations using their mathematical representations.
9. Explain worst, average and best case analysis of algorithms. Can run rime complexity of a comparison-based sorting algorithm be less than N logN?
10. Write Insertion sort algorithm to sort into nonincreasing order. Discuss worst case analysis and give time analysis of the algorithm.
11. With neat figure give Merge sort operations on the array A={3,41,52,26,38,57,9,49} Give time analysis of Merge sort algorithm.
12. Procedure RANDOMIZE-IN-PLACE computes a uniform random permutation. Discuss loop invariant for above statement.

**Unit II**

1. How would you modify Strassen’s algorithm to multiply n x n matrices in which n is not an exact power of 2? Give time analysis of this algorithm.
2. In HIRE-ASSISTANT, assuming that candidates are presented in a random order, what is the probability that you hire exactly twice. Explain your answer.
3. How would you modify Strassen’s algorithm to multiply n x n matrices in which n is not an exact power of 2? Show that the resulting algorithm runs in time Θ (nlg7)
4. Procedure RANDOMIZE-IN-PLACE computes a uniform random permutation.
5. Discuss loop invariant for above statement.
6. Find an optimal parenthesization of a matrix-chain product whose sequence of

Dimensions is {5, 10, 3, 12, 5, 50, 6}.

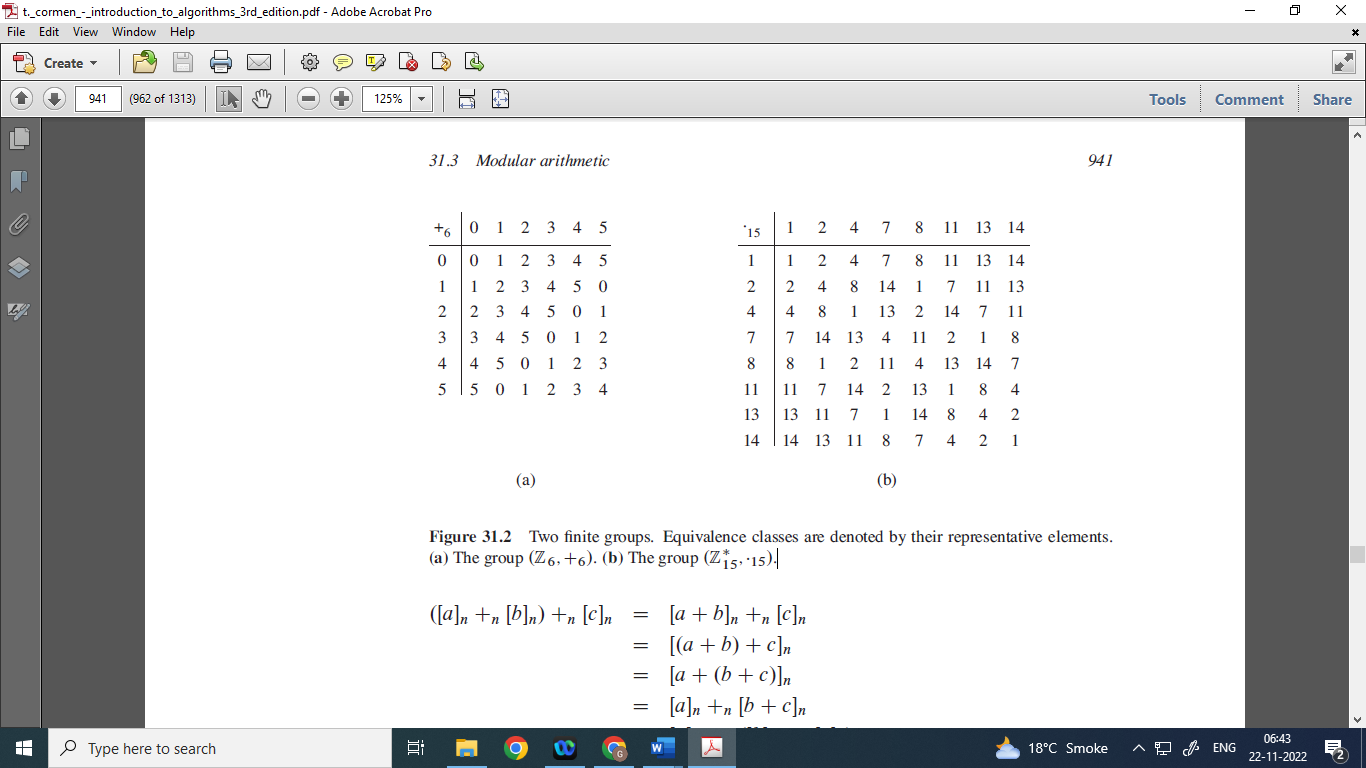
1. What are the indicator random variables? What are their applications?
2. How would you modify Strassen’s algorithm to multiply n x n matrices in which n is not an exact power of 2? Show that the resulting algorithm runs in time Θ (nlg7)
3. Find an optimal parenthesization of a matrix-chain product whose sequence of Dimensions is {5, 10, 3, 12, 5, 50, 6}.
4. How to analyse complexity of recurrence relation?
5. What are the indicator random variables? What are their applications?
6. Use indicator random variables to compute the expected value of the sum of n dice.
7. Draw recursion tree for solving Tower of Hanoi problem using recursion.

Unit III

1. Explain Rod-Cutting problem. Giving example, explain how to solve the problem.
2. Distinguish between Dynamic programming and Greedy approach for solving problem.
3. When developing a dynamic-programming algorithm, what steps to be followed? Give example.
4. Modify MEMOIZED-CUT-ROD to return not only the value but also the actual solution.
5. Give an adjacency-list representation for a complete binary tree on 7 vertices. Give an equivalent adjacency-matrix representation. Assume that vertices are numbered from 1 to 7 as in a binary heap.
6. Modify MEMOIZED-CUT-ROD to return not only the value but also the actual solution.
7. Give an adjacency-list representation for a complete binary tree on 7 vertices. Give an equivalent adjacency-matrix representation. Assume that vertices are numbered from 1 to 7 as in a binary heap.
8. Suppose that instead of always selecting the first activity to finish, we instead select the last activity to start that is compatible with all previously selected activities.
9. Describe how this approach is a greedy algorithm, and prove that it yields an optimal solution.
10. Give comparison of Kruskal and Prim’s algorithm.
11. What is a minimum spanning tree (MST) algorithm? What are its applications? Discuss any one
12. Algorithm which is used for deriving MST.
13. Discuss any one single source shortest path algorithm giving MST.
14. Given an adjacency-list representation of a directed graph, how long does it take to compute the out-degree of every vertex? How long does it take to compute the in-degrees?

Unit IV

1. Give multithreaded pseudocode for transposing an n x n matrix A in place. Analyse the work, span, and parallelism of this algorithm.
2. Compute the values (d, x, y) with the EXTENDED-EUCLID algorithm where x=899, y=493.
3. Calculate two finite groups



1. Find all solutions to the equations x ≡ 4 (mod 5) and x ≡ 5 (mod 11)
2. What is multithreading? How merge sort algorithm can be implemented using multithreading?
3. Giving example explain how modular linear equations can be solved?
4. Give multithreaded pseudocode for transposing an n x n matrix A in place. Analyse the work, span, and parallelism of this algorithm.
5. Explain multithreaded matrix multiplication algorithm. Discuss its time complexity.
6. When a determinacy race occurs in multithreaded algorithms? Give example.
7. Prove that for all integers a, k, and n,

gcd(a,n) = gcd(a + kn, n)

1. Give multithreaded pseudocode for transposing an n x n matrix A in place. Analyse the work, span, and parallelism of this algorithm.
2. Explain multithreaded merge sort algorithm? Is it fast algorithm as compared to merge sort algorithm?

Unit V

1. Is the dynamic-programming algorithm for the 0-1 knapsack problem is a polynomial-time algorithm? Explain your answer.
2. Define P, NP, NP Hard and NP Complete problem. Give two examples of each.
3. State the vertex-cover problem. Is it NP-complete problem?
4. Prove that if a > b > 0 and c = a + b, then c mod a = b.
5. Giving example define P, NP, NP Hard and NP Complete problem.
6. Is the dynamic-programming algorithm for the 0-1 knapsack problem is a polynomial-time algorithm? Explain your answer.
7. Is traveling-salesman problem is NP-complete problem? Explain your answer.
8. Give a formal encoding of directed graphs as binary strings using an adjacency matrix representation. Do the same using an adjacency-list representation. Argue that the two representations are polynomially related.
9. State the vertex-cover problem. Is it NP-complete problem?
10. Is the dynamic-programming algorithm for the 0-1 knapsack problem is a polynomial-time algorithm? Explain your answer.
11. Giving example define P, NP, NP Hard and NP Complete problem.
12. Define Vertex Cover Problem. Why Vertex Cover Problem is a known as NP Complete problem?