

C-IV: Discrete Structures

15th January, 2017

S. No.	Topic	[Reference] Chapter/ Section	No. of Lectures
1.	Sets, Relation, Functions	[1] 1.1 to 1.6* [1] 3.1, 3.3, 3.4, 3.7 (excluding Lattices) * [2] 2.3 [2] 5.2 (Only Introduction Section)	10
2.	Permutation and Combination	[1] 2.1 to 2.4	5
3.	Graphs	[2] Chapter 8 excluding 8.6	10
4.	Trees	[2] 9.1 [2] 9.4 (Only Introduction Section)	
5.	Generating Functions	[1] Chapter 8 excluding 8.3, 8.5	5
6.	Recurrence Relation	[1] Chapter 9 upto 9.7 [3] Chapter 4 (excluding proof of Master's Theorem)	13
7.	Growth of Functions	[3] Appendix A [2] 3.1 (excluding Greedy & Halting) [2] 3.2	5
8.	Logic (Propositional & Predicate)	[1] 1.8 to 1.17 *	12

*Given Program Codes in reference [1] not to be done

References:

- [1] C.L. Liu & Mahopatra, Elements of Discrete mathematics, Third Edition, Tata McGraw Hill
- [2] Kenneth Rosen, Discrete Mathematics and Its Applications, Sixth Edition
- [3] T.H. Cormen, C.E. Leiserson, R. L. Rivest, Introduction to algorithms, Prentice Hall on India (second edition)
- [4] J. P. Trembley, R. Manohar, Discrete Mathematical Structures with application to computer science, Tata McGraw Hill.
- [5] M. O. Albertson and J. P. Hutchinson, Discrete Mathematics with Algorithms

B.Sc.(H) CBES Syllabus.

COMPUTER SCIENCE (C-IV):

Discrete Structures

Total Marks: 100

Total Marks: 150

Theory: 75

Practical: 50

Internal Assessment: 25

Introduction: (15 Lectures)

Sets - finite and Infinite sets, uncountable Infinite Sets; functions, relations, Properties of Binary Relations, Closure, Partial Ordering Relations; counting - Pigeonhole Principle, Permutation and Combination; Mathematical Induction, Principle of Inclusion and Exclusion.

Growth of Functions: (8 Lectures)

Asymptotic Notations, Summation formulas and properties, Bounding Summations, approximation by Integrals

Recurrences: (10 Lectures)

Recurrence Relations, generating functions, Linear Recurrence Relations with constant coefficients and their solution, Substitution Method, Recurrence Trees, Master Theorem

Graph Theory (15 Lectures)

Basic Terminology, Models and Types. multigraphs and weighted graphs, Graph Representation, Graph Isomorphism, Connectivity, Euler and Hamiltonian Paths and Circuits, Planar Graphs, Graph Coloring, Trees, Basic Terminology and properties of Trees, Introduction to Spanning Trees

Propositional Logic (12 Lectures)

Logical Connectives, Well-formed Formulas, Tautologies, Equivalences. Inference Theory

Recommended Books:

1. C.L. Liu & Mahapatra, Elements of Discrete mathematics, 2nd Sub Edition 1985, Tata McGraw Hill
2. Rosen, Discrete Mathematics and Its Applications, Sixth Edition 2006
3. T.H. Cormen, C.E. Leiserson, R. L. Rivest, Introduction to algorithms, Prentice Hall on India (3rd edition 2009)
4. M. O. Albertson and J. P. Hutchinson, Discrete Mathematics with Algorithms 1988 John Wiley Publication 15
5. J. L. Hein, Discrete Structures, Logic, and Computability, Jones and Bartlett Publishers, 3rd Edition, 2009
6. D.J. Hunter, Essentials of Discrete Mathematics, Jones and Bartlett Publishers, 2008

Unique ppr
cd 2

32341202
11th January, 2017

COMPUTER SCIENCE LAB (C-IV): Discrete Structures Practical Lab

Practical: 60 Lectures

(4 Practicals per week (each in group of 10 to 15))

Note: From a computer science perspective, for better understanding of important discrete structures topics like Graphs, students must be taught from an implementation point of view. For the programs on recursion, students must print the intermediate steps, if possible. The computer language used for implementation would be C++.

1. Write a Program to create a SET A and determine the cardinality of SET for an input array of elements (repetition allowed) and perform the following operations on the SET:
 - a) ismember (a, A): check whether an element belongs to set or not and return value as true/false.
 - b) powerset(A): list all the elements of power set of A.
2. Create a class SET and take two sets as input from user to perform following SET Operations:
 - a) Subset: Check whether one set is a subset of other or not.
 - b) Union and Intersection of two Sets.
 - c) Complement: Assume Universal Set as per the input elements from the user.
 - d) Set Difference and Symmetric Difference between two SETS
 - e) Cartesian Product of Sets.
3. Create a class RELATION, use Matrix notation to represent a relation. Include functions to check if a relation is reflexive, Symmetric, Anti-symmetric and Transitive. Write a Program to use this class.
4. Use the functions defined in Ques 3 to find check whether the given relation is:
 - a) Equivalent, or
 - b) Partial Order relation, or
 - c) None
5. Write a Program to generate the Fibonacci Series using recursion.
6. Write a Program to implement Tower of Hanoi using recursion. $T_n = 2T_{n-1} + 1$
7. Write a Program to implement binary search using recursion.
8. Write a Program to implement Bubble Sort. Find the number of comparisons during each pass and display the intermediate result. Use the observed values to plot a graph to analyse the complexity of algorithm.
9. Write a Program to implement Insertion Sort. Find the number of comparisons during each pass and display the intermediate result. Use the observed values to plot a graph to analyse the complexity of algorithm.

$$F(n) = F(n-1) + F(n-2)$$

sum

10. Write a Program that generates all the permutations of a given set of digits, with or without repetition. (For example, if the given set is {1,2}, the permutations are 12 and 21). (One method is given in Liu)
11. Write a Program to calculate Permutation and Combination for an input value n and r using recursive formula of nC_r and nP_r .
12. For any number n , write a program to list all the solutions of the equation $x_1 + x_2 + x_3 + \dots + x_n = C$, where C is a constant ($C \leq 10$) and $x_1, x_2, x_3, \dots, x_n$ are nonnegative integers using brute force strategy.
13. Write a Program to accept the truth values of variables x and y , and print the truth table of the following logical operations:

a) Conjunction $p \wedge q$	f) Exclusive NOR $p \oplus q$
b) Disjunction $p \vee q$	g) Negation \bar{p}
c) Exclusive OR $p \oplus q$	h) NAND $\overline{p \wedge q}$
d) Conditional $p \rightarrow q$	i) NOR $\overline{p \vee q}$
e) Bi-conditional $p \leftrightarrow q$	
14. Write a program to accept an input n from the user and graphically represent the values of $T(n)$ where n varies from 0 to n for the recurrence relations. For e.g. $T(n) = T(n-1) + n$, $T(0) = 1$, $T(n) = T(n-1) + n^2$, $T(0) = 1$, $T(n) = 2 * T(n)/2 + n$, $T(1) = 1$.
15. Write a Program to store a function (polynomial/exponential), and then evaluate the polynomial. (For example store $f(x) = 4n^3 + 2n + 9$ in an array and for a given value of n , say $n = 5$, evaluate (i.e. compute the value of $f(5)$)).
16. Write a Program to represent Graphs using the Adjacency Matrices and check if it is a complete graph.
17. Write a Program to accept a directed graph G and compute the in-degree and out-degree of each vertex.
18. Given a graph G , Write a Program to find the number of paths of length n between the source and destination entered by the user.
19. Given an adjacency matrix of a graph, write a program to check whether a given set of vertices $\{v_1, v_2, v_3, \dots, v_k\}$ forms an Euler path / Euler Circuit (for circuit assume $v_k = v_1$).
20. Given a full m -ary tree with i internal vertices, Write a Program to find the number of leaf nodes.