Micro-Syllabus and Model Question

Discrete Structure

Course Title: Discrete Structure
Course No: BIT152
Nature of Course: Theory + Lab
Semester: II

Full Marks: 60 + 20+20
Pass Marks: 24 + 8 + 8
Credit hours: 3

Course Description: The course introduces the basic concepts of discrete mathematics such as introductory logic, proofs, sets, relations, functions, counting and probability, with an emphasis on applications in information technology.

Course Objectives: The main objective of the course is to introduce basic concepts of discrete mathematics, understand the concepts of graphs, functions, relations and number theory respectively and explore applications of discrete mathematics in information technology.

Detail Contents:

Unit 1: Logic and Proof Methods (6 Hrs.)	
Propositional Logic (Introduction, Propositions, Logical Connectives/Operators, Precedence of Logical Operators, Translating English Sentences to Propositional Logic)	1 Hr.
Propositional Equivalences (Introduction, Logical Equivalences, Proving Logical Equivalences using Truth Table and Symbolic Derivation)	1 Hr.
Predicate Logics (Introduction, Predicates and Quantifiers, Precedence of Quantifiers, Binding Variables, Negation of Quantified Statements, Translating English Sentence to Logical Expressions, Nested Quantifiers)	1 Hr.
Rules of Inferences (Introduction, Rules of Inference for Propositional Logic, Fallacies, Valid Arguments for Propositional Logic, Rules of Inference for Quantified Statements)	1 Hr.
Proof Methods (Introduction and Terminologies, Direct Proof, Indirect Proof, Vacuous and Trivial Proof, Proof by Contradiction, Exhaustive and Proof by Cases, Proof of Equivalence, Existence and Uniqueness Proofs, Proofs by Counter Example), Mistakes in Proofs	2 Hrs.
Unit 2: Sets, Relations and Functions (7 Hrs.)	
Sets (Definition, Notation; Some Important Sets; Equal Sets; Empty Set; Venn Diagram; Subsets; Size of a Set; Power Sets; Cartesian Product; Set Operations – Union, Intersection, Difference and Complement; Computer Representation of Sets – Complement, Union and Intersection)	2 Hrs.
Functions (Definition and Terminologies; Equal Functions; Real Valued and Integer Valued Functions; Image of Subset of Domain; One-to-One, Onto, and One-to-One Correspondence; Inverse and Composite Functions; Graph of Functions; Ceiling Function, Floor Function, Boolean Function and Exponential Function)	2 Hrs.
Relations (Introduction; Functions as Relations; Relation on a Set; Properties of Relations – Reflexive, Symmetric, Antisymmetric, and Transitive; Combining Relations; n-Ary Relations and Applications – Database and Relations, Operations; Representing Relations using Matrices and Diagraphs; Closure of Relations – Reflexive, Symmetric and Transitive; Equivalence Relations and Classes; Partial Orderings)	3 Hrs.
Unit 3: Induction and Recursion (5 Hrs.)	
Mathematical Induction (Introduction; Proofs by Mathematical Induction; Examples – Proving Summation Formula, Proving Inequalities and Proving Divisibility Results)	2 Hrs.
Strong Induction and Well Ordering (Introduction and Examples of Strong Induction; Proofs using Well Ordering Property)	1 Hr.
Recursive Definitions and Structural Induction (Introduction; Recursively Defined Functions, Sets and Structures; Structural Induction)	1 Hr.
Recursive Algorithms and Proving Correctness of Recursive Algorithms	1 Hr.
Unit 4: Number Theory (6 Hrs.)	
Integers and Division (Division Algorithm; Modular Arithmetic; Arithmetic Modulo m)	1 Hr.

Primes and Greatest Common Division (Primes; Trial Division; Prime Factorization; GCD and LCM; Relatively Prime, Pairwise Relatively Prime; Using Prime Factorization to find GCD and LCM)	1 Hr.
Extended Euclidian Algorithm (Euclidian Algorithm; GCDs as Linear Combinations; Extended Euclidian Algorithm)	1 Hr.
Integers and Algorithms (Integer Representations – Binary, Octal, Hexadecimal, and Conversions; Addition, Multiplication, Division, Modulus Algorithms)	1 Hr.
Applications of Number Theory (Linear Congruencies, Chinese Remainder Theorem, Computer Arithmetic with Large Integers)	1 Hr.
Matrices (Zero-One Matrices, Boolean Matrix Operations – Join, Meet, Product, and Power); Prime Number and its applications	1 Hr.
Unit 5: Counting and Discrete Probability (9 Hrs.)	
Counting (Basics of Counting – Product Rule, Sum Rule and Subtraction Rule; Pigeonhole Principle and Generalized Pigeonhole Principle; Permutations and Combinations; Two Element Subsets and Counting Subsets of a Set; Binomial Coefficients and Identity; Pascal's Identity and Triangle; Generalized Permutations and Combinations – Permutation and Combinations with Repetition, Permutations with Indistinguishable Objects; Generating Permutations and Combinations with Examples)	4 Hrs.
Discrete Probability (Finite Probability; Probabilities of Complements and Unions; Probability Theory – Assigning Probability, Conditional Probability, Independence, Random Variables, Expected Value and Variance, Randomized Algorithms, Probability Calculation in Hashing)	2 Hrs.
Advanced Counting (Recurrence Relations; Solving Recurrence Relations - Homogeneous and Non-Homogeneous equations, Theorems without Proof)	3 Hrs.
Unit 6: Tree and Graphs (11 Hrs.)	
Graphs (Graph Basics; Graph Types – Simple Graph, Multigraph, Pseudograph, Directed Graph and Mixed Graph; Graph Models – Social Networks, Communication Networks, Information Networks, Software Design Applications, Transportation Network, Biological Networks and Tournaments; Graph Terminologies; Subgraphs and Union; Complete Graph, Cycle, Wheel, and n-Cube; Bipartite and Complete Bipartite Graphs; Graph Representation – Adjacency List, Adjacency Matrix and Incidence Matrix; Graph Isomorphism; Connectivity in Graphs – Path, Circuit and Connectedness; Euler and Hamilton Paths and Circuits, Necessary and Sufficient Conditions; Matching Theory; Shortest Path Algorithm (Dijkstra's Algorithm); Travelling Salesman Problem; Graph Coloring and Applications – Map Coloring, Exam Scheduling	7 Hrs.
Trees (Introduction; Rooted Trees; Applications – Binary Search Tree, Decision Tree and Prefix Codes; Tree Traversals – Preorder, Inorder and Postorder; Spanning Trees, Minimum Spanning Trees, and Using Kruskal's Algorithm to find Minimum Spanning Trees	4 Hrs.

Laboratory Works:

The laboratory work includes writing computer programs for different algorithms and concepts in the course including.

- Logic
- Set Operations, relations and functions
- Recursive Algorithms
- Primality Testing, Number Theory Algorithms, Operations on Integers, Boolean Matrix Operations
- Algorithms for Counting
- Algorithms for Tree, Graphs

Text / Reference Books:

1. Kenneth H. Rosen, Discrete mathematics and its applications, Seventh Edition McGraw Hill Publication, 2012.
2. Bernard Kolman, Robert Busby, Sharon C. Ross, Discrete Mathematical Structures, Sixth Edition Pearson Publications, 2015
3. Joe L Mott, Abraham Kandel, Theodore P Baker, Discrete Mathematics for Computer Scientists and Mathematicians, Printice Hall of India, Second Edition, 2008

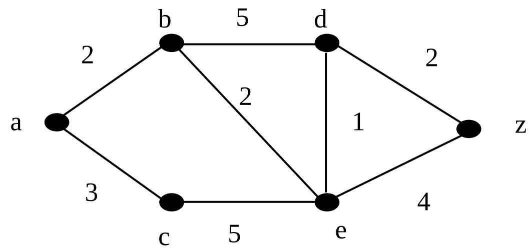
Model Question

Section A

Long Answer Questions

*Attempt any 2 questions. [2*10=20]*

1. Explain mathematical induction. Use mathematical induction to prove that the sum of the first n odd positive integers is n^2. What is recursively defined function? (2 + 6 + 2)
2. Define recurrence relation. What do you mean by linear homogenous recurrence of degree k with constant coefficients? What is the solution of the recurrence relation $a_n = a_{n-1} + a_{n-2}$ with initial conditions and $a_0 = 0$ and $a_1 = 1$. (1 + 2 + 7)
3. What is shortest path finding problem? Use Dijkstra's algorithm to find the length of the shortest path between a and z in the given weighted graphs. (2 + 8)



Section B

Short Answer Questions

*Attempt any 8 questions. [8*5=40]*

4. What do you mean by converse, inverse, and contrapositive? Show that the sentences "if it is hot today then today is Sunday" and "if it is not Sunday then today is not hot" are logically equivalent. (3 + 2)
5. What direct proof? Give a direct proof of the theorem "If n is an odd integer, then n^2 is an odd integer." (1 + 4)
6. Let U = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}. Use bit strings to find the union and intersection of the sets {1, 2, 3, 4, 5} and {1, 3, 5, 7, 9}. (5)
7. Define celling and floor function. Explain Boolean function with example. (2 + 3)
8. How can we represent a relation using directed graph? Draw a directed graph of the relation R = {(1, 1), (1, 3), (2, 1), (2, 3), (2, 4), (3, 1), (3, 2), (4, 1)} on the set {1, 2, 3, 4}. (2 + 3)
9. Explain Euclidean algorithm. Use Euclidean algorithm to find the greatest common divisor of 414 and 662. (2 + 3)
10. Differentiate permutation with combination. What is the next permutation in lexicographic order after 362541? (2 + 3)
11. How can we represent a graph using Adjacency Matrix? Explain. (5)
12. Define spanning tree. Explain minimum spanning tree with example. (1 + 4)