Discrete Structures

Course Title: Discrete Structures
Course No: CSC160
Nature of the Course: Theory + Lab
Semester: II

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

Course Description: The course covers fundamental concepts of discrete structure like introduce logic, proofs, sets, relations, functions, counting, and probability, with an emphasis on applications in computer science.

Course Objectives: The main objective of the course is to introduce basic discrete structures, explore applications of discrete structures in computer science, and understand concepts of Counting, Probability, Relations and Graphs respectively.

Detailed Syllabus

Content Details	Books and References
Unit 1: Basic Discrete Structures (6 Hrs.) 1.1. **Sets:** Sets and Subsets, Power Set, Cartesian Product, Set Operations*(Union, Intersection, Difference, Complement, Symmetric Difference),*Set Identities, Venn Diagram, Inclusion-Exclusion Principle, Computer Representation of Sets. **(2 Hrs)** **1.2. Functions:** Basic Concept,*Sum and Product of Functions*, Injective, *Surjective*,and Bijective Functions, Inverse and Composite Functions, Graph of Functions, Functions for Computer Science (Ceiling Function, Floor Function, Boolean Function, Exponential Function), Fuzzy Sets and Membership Functions, Fuzzy Set Operations. **(3 Hrs)** **1.3. Sequences and Summations**: Basic Concept of Sequences, Geometric and Arithmetic Progression, Single and Double Summation**(1 Hrs)**	*Kenneth H. RosenDiscrete mathematics and its applications* *"Bernard Kolman, Robert Busby, Sharon C. Ross, Discrete Mathematical Structures*"
Unit 2: Integersand Matrices (5 Hrs.) 2.1. **Integers:** Integers and Division,*Division Algorithm*, *Modular Arithmetic,* Primes and Greatest Common Divisor, Extended Euclidean Algorithm, Integers and Algorithms*(Addition, Multiplication, Division & Remainder Algorithms),* Applications of Number Theory (Linear Congruencies, Chinese Remainder Theorem,	*Kenneth H. Rosen Discrete mathematics and its applications*

Computer Arithmetic with Large Integers.**(4 hr)** *2.2.* **Matrices**: Zero-One Matrices, Boolean Matrix Operations*(Join, Product, Boolean Product)****(1 hr)***	
Unit 3: Logic and Proof Methods (6 Hrs.) **3.1. Logic**:Propositional Logic*(Propositions, Compound Propositions, Truth Tables, Representing English Sentences in Propositional Logic)*, Propositional Equivalences, Predicates and Quantifiers,*Precedence and Binding of Quantifiers*, *Predicate Logic Equivalences*, Negation of Quantified Statements, Nested Quantifiers,*Representing English Sentences in Predicate Logic*, Rules of Inferences, Instantiation and Generalization, Proofs by Using Rules of Inferences. **(4 Hrs)** 3.2. **Proof Methods**: Basic Terminologies, Proof Methods (*Direct Proof, Indirect Proof, Proof by Contradiction, Proof by Counter Example, Vacuous and Trivial Proofs, Exhaustive Proofs and Proof by Cases*), Mistakes in Proof.**(2 Hrs)**	*Kenneth H. Rosen Discrete mathematics and its applications*
Unit 4: Induction and Recursion (5 Hrs.) **4.1. Induction:** Mathematical Induction,*Proofs by Using Mathematical Induction*, Strong Induction, *Proofs by Using Strong Induction*, Well Ordering, *Proofs by using Well Ordering*, Induction in General**(3 Hrs)** 4.2. **Recursive Definitions and Structural Induction**:*Recursively defined Functions, Sets, and Structures*, Structural Induction, *Proofs by Using Structural Induction*, Recursive Algorithms, Proving Correctness of Recursive Algorithms**(2 Hrs)**	*Kenneth H. Rosen Discrete mathematics and its applications*

Unit 5: Counting and Discrete Probability (10 Hrs.) **5.1. Counting**: Basics of Counting, *Sum and Product Rule*,*Counting Problems*, Pigeonhole Principle,*Generalized Pigeonhole Principle*, *Applications of Pigeonhole Principle*, Permutations and Combinations, Two Element Subsets, Counting Subsets of a Set, Binomial Coefficients,*Binomial Theorem (without proof)*,*Pascal's Identity and Triangle*, Generalized Permutations and Combinations, *Permutations and Combinations with Repetition*, *Permutations and Combinations with Indistinguishable Objects*, Generating Permutations and Combinations.**(6 Hrs)** 5.2. **Discrete Probability**: Introduction to Discrete Probability, Probability Theory, *Conditional Probability, Independence, random Variable*,*Probabilistic Primility Testing*, Expected Value and Variance,*Concept and Examples of* Randomized Algorithms.**(2 Hrs)** 5.3. **Advanced Counting**: Recurrence Relations, Solving *Linear*Recurrence Relations (Homogeneous and Non-Homogeneous equations, *Theorems without Proof*), Introduction to Divide and Conquer Recurrence Relations.**(2 Hrs)**	*Kenneth H. Rosen Discrete mathematics and its applications*
Unit 6: Relations and Graphs (13 Hrs.) 6.1. **Relations:** Relations and their Properties,*Combining Relations*, N-ary Relations, *Operations on N-ary Relations*, Applications of N-ary Relations, Representing Relations by using *Matrix and Diagraphs*, *Reflexive*, *Transitive and Symmetric*Closure of Relations, Equivalence Relations,*Equivalence Classes*, *Partitions*, Partial Ordering, *Total Ordering*, *Lexicographical ordering*, *lattice*.**(4 Hrs)** 6.2. **Graphs:** Graphs Basics, Graph Types, Graph Models,*Graph Terminologies*, *Simple and Special Graphs*, *Subgraphs*, Graph Representation, Connectivity in Graphs, *Paths*, *Connected Graph and Component*, *Strongly and Weakly Connected Graphs*, *Counting Paths*,Graph Isomorphism,Euler Path and Circuit, Hamiltonian Path and Circuit, Matching Theory, Shortest Path Algorithm (Dijkstra's Algorithm), Travelling Salesman Problem, Graph Coloring, *Applications of Graph Coloring*. (*Theorems in each subtopic must be considered*)**(5 Hrs)** 6.3. **Trees:** Introduction and Applications, Tree Traversals, Spanning Trees, Minimum Spanning Trees (Kruskal's Algorithm)**(2 Hrs)**	*Kenneth H. Rosen Discrete mathematics and its applications* *Joe L Mott, Abraham Kandel, Theodore P Baker, Discrete Mathematics for Computer Scientists and Mathematicians,* *"Bernard Kolman, Robert Busby, Sharon C. Ross, Discrete Mathematical Structures*"

6.4. **Network Flows**: Graph as Models of Flow of Commodities, Flows, SD-Cut, Maximal Flows and Minimal Cuts, The Max Flow-Min Cut Theorem.**(2 Hrs)**	

Laboratory Work (45 Hrs)

The laboratory work consists of implementing the algorithms and concepts discussed in the class. Student should write programs to demonstrate concepts listed below.

Unit 1 (10 Hr)

1. Programs to implement set operations union, intersection, difference, and Cartesian product
2. Programs to implement ceiling and floor functions
3. Programs to implement fuzzy set operations

Unit 2 (10 Hr)

1. Programs to implement Euclidean and Extended Euclidean algorithms
2. Programs to implement binary integer addition, multiplication, and division
3. Programs to implement Boolean matrix operations join, product, and Boolean product
4. Programs to perform operations with large integers by breaking down them into set of small integers

Unit 3 (6 Hr)

1. Programs to generate truth tables of compound propositions
2. Programs to test validity of arguments by using truth tables

Unit 4 (2 Hr)

1. Programs to compute a^n, b^n mod m, linear search etc by using recursion

Unit 5 (7 Hr)

1. Programs to generate permutations and combinations
2. Programs to implements some probabilistic and randomized algorithms

Unit 6 (10 Hr)

1. Programs for representing relations, testing its properties, and testing equivalence
2. Programs to represent graphs, finding shortest path, and generating minimum spanning trees

Model Question

Section A

Long Answer Questions

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

1. Why breaking down of large integer into set of small integers is preferred while performing integer arithmetic? Find sum of numbers 123,684 and 413,456 by representing the numbers as 4-tuple by using reminders modulo of pair-wise relatively prime numbers less than 100. {2+8}
2. Define linear homogeneous recurrence relation. Solve the recurrence relation $a_n=a_{n/2}+n+1$, with $a_1=1$.Also discuss about probabilistic primility testing with example. {2+4+4}
3. How Zero-one matrix and diagraphs can be used to represent a relation? Explain the process of identifying whether the graph is reflexive, symmetric, or anti-symmetric by using matrix or diagraph with suitable example. {4+6}

Section B

Short Answer Questions

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

4. Prove that $\overline{A \cap B} = A \cup B$ by using set builder notation. How sets are represented by using bit string? Why it is preferred over unordered representation of sets? {3+2}
5. How can you relate domain and co-domain of functions with functions in programming language? Discuss composite and inverse of function with suitable examples. {2+3}
6. State Euclidean and extended Euclidean theorem. Write down extended Euclidean algorithm and illustrate it with example. {1+4}
7. State and prove generalized pigeonhole principle? How many cards should be selected from a deck of 52 cards to guarantee at least three cards of same suit? {2.5+ 2.5}
8. Represent the argument "If it does not rain or if is not foggy then the sailing race will be held and the lifesaving demonstration will go on. If sailing race is held then trophy will be awarded. The trophy was not awarded. Therefore it not rained" in propositional logic and prove the conclusion by using rules of inferences. {2+3}
9. Discuss common mistakes in proof briefly. Show that n is even if n^3+5 is odd by using indirect proof. {2+3}
10. How mathematical induction differs from strong induction? Prove that $1^2 + 2^2 + 3^2 + \cdots \ldots \ldots n^2 = n(n+1)(2n+1)/6$ by using strong induction. {1+4}
11. Write down recursive algorithm for computing a^n. Argue that your algorithm is correct by using induction. {2.5+2.5}
12. What is meant by chromatic number? How can you use graph coloring to schedule exams? Justify by using 10 subjects assuming that the pairs {(1,2), (1,5), (1,8), (2,4), (2,9), (2,7), (3,6), (3,7), (3,10), (4,8), (4,3), (4,10), (5,6), (5,7)} of subjects have common students. {1+4}

Guideline for Question Setter

- → There must be at least one question from each unit
- → There should not be more than one question from a sub-unit