

**Tribhuvan University**  
**Institute of Science and Technology**  
**Micro – Syllabus**

**Course Title: Discrete Structure**

**Course no:** CSC-152

Full Marks: 80+20

**Credit hours:** 3

Pass Marks: 32+8

**Nature of course:** Theory (3 Hrs)

**Course Synopsis:**

This course contains the fundamental concepts of logic, reasoning and algorithms.

**Goal:** After completing this course, the target student will gain knowledge in discrete mathematics and finite state automata in an algorithmic approach. It helps the target student in gaining fundamental and conceptual clarity in the area of logic, reasoning, algorithms, recurrence relation, and graph theory.

**Course Contents:**

**Unit1. Logic, Induction and reasoning**

(12 hrs)

**1.1 Logic**

Propositions, truth value, propositional logic, compound propositions, logical connectives (negation, conjunction, disjunction, exclusive OR, implication converse, contrapositive and biconditional, truth table for each), translation of English statements into logical expressions, application of logic in Boolean searches.

Contradiction, tautology, contingency, logical equivalences, laws of logical equivalences (identity, domination, laws, double negation, commutative, associative, distributive, absorption and DeMorgan's), proving logical equivalences by using truth tables and laws of logical equivalence, dual of compound proposition.

Predicate logic, quantifiers and quantification, universal and existential quantification, translation of quantified expressions into English sentences and vice versa, bounded and free variables, negations of quantified expression.

Examples related to each concept.

**1.2 Induction and Reasoning**

Rules of inferences (addition, simplification, conjunction, modus ponens, modus tollens, hypothetical syllogism, disjunctive syllogism and resolution), proving the validity of arguments, resolution, fallacies (affirming the conclusion, denying the hypothesis and circular reasoning), quantified statements, rules of inference for quantified statements (universal instantiation, universal generalization, existential instantiation and existential generalization).

Methods of proving theorems (trivial proof, vacuous proof, direct proof, indirect proof, proof by contradiction, proof by cases, existence proof, proof by counter example, proof by weak and strong inductions, recursive induction.

Examples related to each concept.

## Unit 2 : Finite state Automata

(10 hrs)

### 2.1 Deterministic Finite Automata

Concept of sequential circuit design, use of Finite State Machine in sequential circuit design, Finite State Machine with output and without output, formal definition of Deterministic Finite Automata, concatenation, Kleene closure, application of DFA, language of DFA, examples of languages accepted by DFA, related problems.

### 2.2 Non- Deterministic Finite Automata

Formal definition of non-deterministic finite automata, difference between NFA and DFA, language of NFA, examples of languages accepted by NFA,

Related problems in each case.

### 2.3 Regular Expressions

Formal definition of regular expressions, regular languages, similarities in regular expressions and automata, application of regular expressions, related exercises.

### 2.4 Languages and Grammars

Natural language and formal language, syntax and semantics, Phrase structure grammar, derivation, Language generated by a grammar, Types of phrase structure grammar, Related problems

## Unit3: Recurrence Relations

(8 hrs)

### 3.1 Recurrence Relation

Introduction to recurrence relations, definition of recurrence relations, Fibonacci numbers, recursively defined sets and structures, basic concepts of combinatorics (sum and product rules, the Pigeonhole principle, permutation, combination, binomial coefficients), initial condition, modeling problems with recurrence relation, related exercises.

### 3.2 Solution of Recurrence Relation

Definition and examples of linear homogeneous recurrence relation. Verification of following theorems without proof:

**Theorem 1 (page 414):** Let  $c_1$  and  $c_2$  be two real numbers. Suppose  $r^2 - c_1r - c_2 = 0$  has two distinct roots  $r_1$  and  $r_2$ . Then the sequence  $\{a_n\}$  is a solution of the recurrence relation  $a_n = c_1 a_{n-1} + c_2 a_{n-2}$  if and only if  $a_n = \alpha_1 r_1^n + \alpha_2 r_2^n$  for  $n = 0, 1, 2, \dots$  where  $\alpha_1$  and  $\alpha_2$  are constants.

**Theorem 2 (page 416):** Let  $c_1$  and  $c_2$  be two real numbers with  $c_2 \neq 0$ . Suppose  $r^2 - c_1r - c_2 = 0$  has only one root  $r_0$ . Then the sequence  $\{a_n\}$  is a solution of the recurrence relation  $a_n = c_1 a_{n-1} + c_2 a_{n-2}$  if and only if  $a_n = \alpha_1 r_0^n + \alpha_2 n r_0^n$ , for  $n = 0, 1, 2, \dots$ , where  $\alpha_1$  and  $\alpha_2$  are constants.

**Theorem 3 (page 417):** Let  $c_1$  and  $c_2, \dots, c_k$  be  $k$  real numbers. Suppose that the characteristic equation  $r^k - c_1 r^{k-1} - \dots - c_k = 0$  has  $k$  distinct roots  $r_1, r_2, \dots, r_k$ . Then the sequence  $\{a_n\}$  is a solution of the recurrence relation  $a_n = c_1 a_{n-1} + c_2 a_{n-2} + \dots + c_k a_{n-k}$  if and only if  $a_n = \alpha_1 r_1^n + \alpha_2 r_2^n + \dots + \alpha_k r_k^n$ , for  $n = 0, 1, 2, \dots$ , where  $\alpha_1, \alpha_2, \dots, \alpha_k$  are constants.

**Theorem 4 (page 418):** Let  $c_1$  and  $c_2, \dots, c_k$  be  $k$  real numbers. Suppose that the characteristic equation  $r^k - c_1 r^{k-1} - \dots - c_k = 0$  has  $t$  distinct roots  $r_1, r_2, \dots, r_t$  with multiplicities  $m_1, m_2, m_3, \dots, m_t$  respectively. Then a sequence  $\{a_n\}$  is a solution of the recurrence relation  $a_n = c_1 a_{n-1} + c_2 a_{n-2} + \dots + c_k a_{n-k}$  if and only if  $a_n = (\alpha_{1,0} + \alpha_{1,1}n + \dots + \alpha_{1,m_1-1}n^{m_1-1})r_1^n + (\alpha_{2,0} + \alpha_{2,1}n + \dots + \alpha_{2,m_2-1}n^{m_2-1})r_2^n + \dots + (\alpha_{t,0} + \alpha_{t,1}n + \dots + \alpha_{t,m_t-1}n^{m_t-1})r_t^n$  for  $n = 0, 1, 2, \dots$ , where  $\alpha_{i,j}$  are constants for  $1 \leq i \leq t$  and  $0 \leq j \leq m_i - 1$ .

Definition and examples of linear non homogeneous, associated homogeneous recurrence relations, verification of following theorems without proof:

**Theorem 5 (page 420):** If  $\{a_n^{(p)}\}$  is a particular solution of the non-homogeneous recurrence relation with constant coefficients  $a_n = c_1 a_{n-1} + c_2 a_{n-2} + \dots + c_k a_{n-k} + F(n)$ , then every solution of the form  $\{a_n^{(p)} + a_n^{(h)}\}$ , where  $\{a_n^{(h)}\}$  is solution of associated homogeneous recurrence relation  $a_n = c_1 a_{n-1} + c_2 a_{n-2} + \dots + c_k a_{n-k}$

**Theorem 6 (page 421):** Suppose that  $\{a_n\}$  satisfies the linear homogeneous recurrence relation  $a_n = c_1 a_{n-1} + c_2 a_{n-2} + \dots + c_k a_{n-k} + F(n)$ , where  $c_1, c_2, \dots, c_k$  are real numbers and  $F(n) = (b_t n^t + b_{t-1} n^{t-1} + \dots + b_1 n + b_0) s^n$  where  $b_0, b_1, \dots, b_t$  and  $s$  are real numbers. When  $s$  is not root of characteristic equation of the associated linear homogeneous recurrence relation, there is a particular solution of the form  $(p_t n^t + p_{t-1} n^{t-1} + \dots + p_1 n + p_0) s^n$ , where  $s$  is root of this characteristic equation and its multiplicity is  $m$ , there is a particular solution of the form  $n^m (p_t n^t + p_{t-1} n^{t-1} + \dots + p_1 n + p_0) s^n$ .

Solution of recurrence relation to algorithm analysis, divide and conquer relations, partial order relation. Examples illustrating each.

## Unit 4: Graph Theory

(15 hrs)

### 4.1 Introduction

Definition of directed and undirected graphs, simple and multigraphs, pseudographs, edges, vertices, isolated and pendant vertices, adjacent vertices, incident edge, adjacent and incident matrices representations of graphs, adjacency lists, in-degree, out-degree of a vertex, regular graph, Graph isomorphism.

Statement of *Handshaking Theorem* (Theorem 1, page 546), proof of Theorem 2, page 547, statement of Theorem 3, page 548, verifications of the theorem by examples.

Examples of some special graphs (round-robin tournaments, the complete graph  $K_n$  on  $n$  vertices, the cycles  $C_n$ , the wheel  $W_n$ ,  $n$ -cube  $Q_n$ ), bipartite graphs and complete bipartite graphs  $K_{m,n}$ , applications of graphs on local-area networks, subgraph of a graph and union of graphs. Example related to all concepts.

## 4.2 Graph Connectivity

Definitions and examples of walk and path and circuit, cut vertices/Edges, cut sets, connectedness in undirected and directed graphs, weakly and strongly connected graphs, underlying graphs, connected components. Proof of Theorem 1, page 570 (There is a simple path between every pair of distinct vertices of a connected undirected graphs), related exercises.

## 4.3 Euler and Hamiltonian Graphs

Definitions and examples of Euler paths and circuits, multigraph model of the town of Königsberg, proof of Theorem 1, page 581 (A connected autograph has an Euler circuit if and only if each of its vertices has even degree), Algorithm 1 (page 581), *Constructing Euler Circuits*, examples illustrating the algorithm, proofs of Theorem 3, page 586 (a connected multigraph has an Euler path but not an Euler circuit if and only if exactly two of its vertices has odd degree), related exercises.

Definition of Hamiltonian paths and circuits, examples illustrating existence and nonexistence of Hamiltonian circuit,  $K_n$  has Hamiltonian circuit whenever  $n > 2$ , application of *Dirac's Theorem* and *Ore's Theorem*, application of Hamiltonian circuit to travelling salesman problem, related exercises.

## 4.4 The Shortest Path Algorithm

Definition of weighted graph, the *Shortest Path Algorithm of Dijkstra*, examples to illustrate the algorithm, idea of travelling salesman problem in connection to the shortest path algorithm, computational difficulty of the travelling salesman problem, related exercises.

## 4.5 Planar Graphs

Definitions and examples of planar graphs, statements of the following results without proof.

**Theorem 1** (page 606) . Let  $G$  be a connected planar simple graph with  $e$  edges and  $v$  vertices. Let  $r$  be the number of regions in a planar representation of  $G$ . Then  $r = e - v + 2$ .

**Corollary 1.** If  $G$  is connected planar simple graph with  $e$  edges and  $v$  vertices. where  $v \geq 3$ , then  $e \leq 3v - 6$ .

**Corollary 2.** If  $G$  has a connected planar simple graph, then it has a vertex of degree not exceeding five.

**Corollary 3.** If  $G$  is connected planar simple graph with  $e$  edges and  $v$  vertices where  $v \geq 3$  and no circuits of length 3, then  $e \leq 2v - 4$ .

Homeomorphism, Related examples illustrating the results in every case.

#### 4.6 Graph Coloring

Basic concept, definition of Chromatic number, the statement of *Four Color Theorem* (The chromatic number of a graph is no greater than four), applications of graph colorings to scheduling final exams, illustrating the concepts by examples.

#### 4.7 Trees and Spanning Trees

Definitions and examples of tree and forest, parent, child, rooted tree, ordered rooted tree ancestor, descendent, sibling, level/depth of node, depth of a tree, internal nodes, leaf nodes, subtree, binary tree. Definitions of *m-ary* and full *m-ary* tree, trees as models (computer file systems, tree-connected parallel-processors), related examples.

Proofs of Theorem 1, page 633 (An undirected graph is tree if and only if there is a unique simple path between any two of its vertices ) and Theorem 2, page 638 (A tree with  $n$  vertices has  $n-1$  edges). Statement of Theorem 3, page 639, (A full  $m$ -ary tree with  $i$  internal vertices contains  $n = mi+1$  vertices). Verifications of the result by examples.

Applications of trees (Algorithm 1, page 646, Binary Search Tree Algorithm), decision trees, Prefix codes, related examples.

Definition of spanning trees, proof of Theorem 1, page 676 (a simple graph is connected if and only if it has spanning tree), finding minimal spanning tree by using Kruskal's algorithm, related exercise.

#### 4.8 Network Flow Problems

Concept of network flows, proof of Maxflow and Mincut theorem, verification of the algorithms by examples.

#### Text Books:

1. Kenneth H. Rosen, Discrete Mathematical Structures with Application to Computer Science, WCB/McGraw Hill.
2. Joe L. Mott, Abrahan Kandel and Theodore P. Baker, Discrete Mathematics for Computer Scientists and Mathematicians, Prentice-Hall of India.

#### Reference Books:

1. R. Johnsonbaugh, Discrete mathematics, Prentical Hall Inc.
2. G. Chartand, B.R. Oller Mann, Applied and Algorithmic Graph Theory, McGraw Hill.
3. G. Birkhoff, T.C. Bartee, Modern Applied Algebra, CBS Publishers.

#### Remarks:

1. Theory and practice should be done side by side.
2. Theory classes 4hrs and tutorial classes 2 hrs per week.
3. Recommended to use Mathematica/Matlab/Maple for testing selected exercises.