COURSE NUMBER: MACM 201

CREDITS: 3

COURSE TITLE: Discrete Mathematics II

PREREQUISITES: MACM 101 with a minimum grade of "C"

Total Hours: 52

COURSE DESCRIPTION: A continuation of MACM 101. Topics covered include inclusion-exclusion, generating functions and recurrence relations, graph theory and trees.

LEARNING OUTCOMES:

Upon successful completion of this course, you should be able to:

1. Inclusion-Exclusion:
    a) Apply the inclusion-exclusion principle to problems with more than two sets.
    b) Use the principle of inclusion-exclusion to solve counting problems modeled after the problem of finding the number of integer solutions of a linear equation with constraints.
    c) Solve counting problems modeled after the number of onto functions from one finite set to another.
    d) Count the number of derangements of a set and solve counting problems based on this principle.
2. Generating Functions:
    a) Derive generating functions for a sequence.
    b) Use ordinary and exponential generating functions to solve counting problems.
3. Recurrence Relations:
    a) Develop a recurrence relation to model a problem.
    b) Solve recurrence relations iteratively.
    c) Solve linear homogenous recurrence relations with constant coefficients of second order.
    d) Verify solutions to linear nonhomogenous recurrence relations.
    e) Determine the big-O of divide-and-conquer recurrence algorithms such as the binary search.
4. Graphs:
    a) Represent a graph as an adjacency matrix of incidence matrix.
    b) Determine whether a pair of graphs is isomorphic.

c) Demonstrate an understanding of connected, simple path, weighted graph, circuit, subgraph, complement, cute verticles, cut edges and degree of a vertex.

d) Determine whether or not a graph has an Euler path or circuit.

e) Determine whether or not a graph has a Hamiltonian path or cycle.

f) Model and solve problems with a graph

g) Use Dijkstra's algorithm to find the shortest path between pairs of verticles in a weighted graph.

h) Determine whether or not a graph is planar.

i) Understand the chromatic polynomial of a graph.

5. Trees:

a) Demonstrate an understanding of the terminology of rooted trees and m-ary trees.

b) Form a binary search tree using a recursive algorithm and analyze the algorithm.

c) Model a problem using a decision tree.

d) Construct a binary tree which represents a given prefix coding scheme.

e) Find the word represented by a bit string given a coding scheme.

f) Demonstrate an understanding of a Huffman tree.

g) Demonstrate an understanding of three algorithmsfor traversing all verticles of an ordered rooted tree (preorder, inorder and postorder traversal).

h) Represent a complicated expression using a binary tree and write the expression in prefix, postfix or infix notation.

i) Evaluate a prefix, postfix or infix expression

j) Demonstrate an understanding of several sorting algorithms and their complexity (bubble sort, merge sort, selection sort and quick sort).

k) Find all the non-isomorphic spanning trees of a graph.

l) Use a depth-first or breadth-first search to produce a spanning tree.

m) Use Prim's or Kruskal's algorithm to construct a minimum spanning tree for a weighted graph

COURSE CONTENT:

Week	**Topic**
Week 1	Graph Theory; Definitions; Subgraphs; Complements
Week 2	Graph Isomorphism Vertex Degree: Euler Trails and Circuits; Planar Graphs
Week 3	Hamilton Paths and Cycles; Graphs Coloring (optional)

Week 4	Advanced Probability; Review of Finite Probability; Conditional Probability
Week 5	Inclusion-Exclusion; The Principle of Inclusion-Exclusion
Week 6	Generalized Inclusion-Exclusion; Derangements
Week 7	Advanced Enumeration; Introduction to Generating Functions; Calculational Techniques;
Week 8	Partitions of Integers; Recurrence Relations; First-Order Linear
Week 9	Recurrence Relations; Second-Order Linear Homogenous
Week 10	Recurrence Relations with Constant Coefficients; Nonhomogenous Recurrence Relations
Week 11	The Method of Generating Functionsl Optimization and Matching; Review of Trees
Week 12	Dijkstra's Shortest-Path Algorithm; Minimum Spanning Trees: Kruskal's
Week 13	Prim's Algorithms; Matching Theory (optional)