Spring 2016

Math 301 – Introduction to Discrete Mathematics (3)

Symbolic logic, sets and relations, algorithms, mathematical induction, recurrence relations, trees and other graphs.

Pre: One semester of calculus from mathematics department; or consent. Recommended: one semester programming.

Approximate timeline:

1. Fundamentals of Logic and Sets

Logic, Propositional Equivalences, Predicates and Quantifiers, Sets, Set Operations, Functions, Sequences.

2. Algorithms

Algorithms, Complexity, Integer Algorithms, Matrices.

3. Mathematical Reasoning

Proofs, Mathematical Induction, Recursive Definitions and Algorithms.

4. Counting

Rules of Sum and Product, Pigeonhole Principle, Permutations, Combinations, Discrete Probability.

5. Advanced Counting

Recurrence Relations, Divide-and-conquer Relations, Inclusion-Exclusion.

6. Relation Algorithms

Relations, n-ary Relations, Representing Relations, Closures, Equivalence Relations, Partial Orders.

7. Graphs

Graphs, Representing Graphs, Graph Isomorphism, Connectivity, Euler and Hamilton Paths, Shortest Paths, Planar Graphs, Graph Coloring.

8. Trees

Trees, Applications, Tree Traversal, Trees and Sorting, Spanning Trees, Minimal Spanning Trees.

Possible texts: K. Rosen, *Discrete Mathematics and Its Applications*, Random House.
R. P. Grimaldi, *Discrete and Combinatorial Mathematics*,
Addison-Wesley.
Lovasz, Pelikan, Vesztergombi, *Discrete Mathematics and Beyond*, Springer.

Course Objectives: Upon successful completion, the student will have a working knowledge of discrete mathematics. The student will be able to apply this to problems involving counting, inductive reasoning and analysis of algorithmic complexity.

Program Objectives: Discrete mathematics provides a foundation for understanding advanced computer science. In particular the analysis of program correctness, termination and complexity.