Shih Chien University

STP Program (Dec 22-Jan 16)

MATH 410 Discrete Mathematics for Computer Science

Course Outline

Course Code: MATH 410

Instructor: Prof. Vadim Olshevsky

Home Institution: University of Connecticut

Office Hours: TBA and by appointment

Email: olshevsky@gmail.com

Credits: 4

Class Hours:
The course comprises a total of 144 instructional hours, consisting of 50 hours of lectures, 30 hours of professor office hours, 20 hours of TA-led discussion sessions, 10 hours of review sessions, and 34 hours of supplementary classes.

Course Description:

This course provides an introduction to discrete mathematics, laying the foundational concepts essential for modern computer science. It emphasizes not only the acquisition of mathematical tools and techniques but also the development of logical and analytical thinking skills crucial for problem-solving.
Students will learn formal proof techniques based on mathematical logic and understand how these methods underpin algorithm design. The course also explores combinatorial analysis (such as counting and enumeration) to address computational problems and evaluate algorithmic efficiency. In addition, students will study discrete data structures that implement mathematical models frequently used in algorithm development.

Topics Covered

This course will cover the following key concepts in mathematical modeling and discrete structures:

1. Sets
2. Logic
3. Number Theory and Cryptography
4. Proof Techniques
5. Sequences and Functions
6. Relations and Relational Databases
7. Graph Theory and Trees
8. Probability
9. Combinatorics
10. Algorithm Efficiency

Learning Outcomes

By the end of this course, successful students will be able to:

- Translate everyday language propositions into symbolic logic and evaluate their truth values.
- Manipulate, simplify, and relate symbolic logical statements.
- Explain and apply a variety of proof techniques, including mathematical induction, proof by contradiction, contrapositive reasoning, and the pigeonhole principle.
- Determine appropriate proof strategies for different types of problems.
- Define mathematical sets, set operations, and functions, and connect these concepts to discrete data structures.
- Apply counting techniques—such as permutations, combinations, and binomial coefficients—to solve computational and combinatorial problems.
- Explain and apply key concepts of discrete probability and understand their relevance to algorithm analysis.
- Use discrete data structures (such as graphs and trees) to model, analyze, and solve algorithmic and computational problems.

Modality

This course is delivered **online asynchronously**. All lectures are pre-recorded and available from the first day of class. Homework assignments are completed online via **WebAssign**. Two proctored exams (midterm and final, each 1 hour 15 minutes) will be conducted via **Zoom**. Students must use a webcam with a wide view that clearly shows both themselves and their workspace.

Required Course Materials:
Discrete Mathematics with Applications (5th Edition) by Susanna S. Epp. Your instructor will provide a WebAssign registration link, which will also include options for purchasing access to the digital textbook.

Homework: There will be online WebAssign homework assignments for each section of the text. Each assignment will be made available several days before the section is covered in class. The due date for each assignment will be set by your instructor and will generally be two or three days after the material is covered in class. You will get five attempts for each question.

Grading & Evaluation:

Attendance and participation:	10%
Homework:	30%
Midterm:	30%
Final:	30%

Grading System (1 ~ 100):

A+ : 96 - 100	A : 91 - 95
B+ : 86 - 90	B : 81 - 85
C+ : 76 - 80	C : 71 - 75
D+ : 66 - 70	D : 60 - 65
F : 0 - 59	
Pa : Pass	Fa : Fail

Brief Course Schedule
The following outline is tentative and may be adjusted based on the progress and needs of the class.

Week 1
Introduction to Set Theory
Propositional Logic and Truth Tables

Logical Equivalence and Predicate Logic
Direct Proof Methods

Week 2
Number Theory
RSA Encryption and Decryption
Proof by Contrapositive and by Contradiction
Sequences and Mathematical Induction
Review of Set Theory
Boolean Algebra and Russell's Paradox
Cryptographic Hash Functions

Week 3
Relations and Graphs
Relations on Sets and Relational Databases
Modular Arithmetic and Cryptography
PERT and CPM Network Models
Introduction to Counting and Probability

Week 4
Graph and Tree Theory
Binary Search Trees and Minimum Spanning Trees
Kruskal's, Prim's, and Dijkstra's Algorithms
Algorithm Efficiency and Divide-and-Conquer Strategies

Detailed Course Schedule:

Week	Chapter	Topic
	2	The logic of compound statements. Logical form and logical equivalence. Conditional statements. Valid and invalid arguments. Application: Number systems.
1		Application: digital logic circuits. The logic of quantified statements. Predicates and quantified statements.
	3	The logic of quantified statements. Predicates and quantified statements. Statement with multiple quantifiers. Arguments with quantified statements

		Elementary number theory. Direct proofs and proofs by a counterexample. Generalizing from generic particular. Example: properties of rational numbers. Example: divisibility and factorizations of integers.
	4	Division into cases and the Quotient-Remainder theorem. *Div* and *mod*. Proof by contradiction. Proof by contraposition. Irrationality of sqrt{2}. Infinite number of primes. The Euclidean algorithm. Sequences. Geometric sequence. Principle of mathematical induction. Inductive reasoning. Proving divisibility properties, proving inequalities.
2		Trominoes and other applications. Strong mathematical induction and the well-ordering principle. Defining sequences recursively. The method of iteration. Second order linear homogeneous recurrence relations with constant coefficients. The distinct roots and the single root cases. Recursively defined sets and Boolean expressions.
	5	Sets, subsets, Venn diagrams, empty set, partition of sets, power set. Set identities. sets properties. Proving basic. Proving that a set is an empty set.
		Disproving an alleged set property. The number of subsets of a set. “Algebraic proofs” of set identities. Boolean algebras. Russel’s paradox.
	6	Functions defined on a set. Boolean functions. One-to-one functions. Hash and cryptographic hush functions. Onto functions. Inverse functions. Composition of functions. Composition of one-to-one and onto functions.
3		Directed graph of a relation. N-ary relations and relational databases. Reflexive, symmetric and transitive properties. The transitive closure. Equivalence relations. Equivalence classes. Relations on sets. The inverse of a relation.
	7	Modular arithmetic and cryptography. Modular arithmetic and Euclidean algorithm. RSA cryptography. Fermat’s little theorem. Antisymmetry, partial order relations. Hasse diagrams, partially and totally ordered sets. Application: PERT and CPM.
		Intro to probability. Events, sample spaces. Basic principles of counting. Possibility trees. The multiplication rule. Permutations. The addition rule, the difference rule. The

		inclusion/exclusion rule. The pigeonhole principle. Combinations.
	8	R-combinations and multisets. Pascal's formula and the binomial theorem. Probability axions. Expected value. Conditional probability, Bayes' formula and independent events.
4		Trees: examples and basic properties. Rooted, binary trees. Binary search trees. Minimum spanning trees. Kruskal's algorithm. Prim's algorithm. Dijkstra's shortest path algorithm.
	9	Analysis of algorithm efficiency. Real functions and their graphs. Big- O, big-Omega and big-Theta notations. Orders of power functions. Measuring the efficiency of an algorithm. Computing orders of simple algorithms.
		Graphs of exponential and logarithmic functions. Application: number of bits to represent an integer. Exponential and logarithmic orders. Binary search, divide and conquer algorithms. Efficiency of binary search and merge sort. Tractable and intractable problems.

Student Responsibilities and Expectations

The primary course content will be delivered through lectures. Students are expected to keep up with the material as it is presented throughout the term. Regular attendance at lectures and discussion sessions is strongly encouraged, along with consistent effort on homework assignments.
If a student misses a class, it is their responsibility to obtain the lecture notes and review the missed material. Students are encouraged to actively ask questions and seek additional help whenever needed to ensure they remain on track with the course.