# Shih Chien University

## STP Program (June 02-July 04)

## 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**

**Email: olshevsky@gmail.com**

**Credits: 4**

**Class Hours:**
This course will have 144 class hours, including 50 lecture hours, professor 30 office hours, 20-hour TA discussion sessions, 10-hour review sessions, 34-hour extra classes.

**Prerequisites:** N/A

### Course Description:

This course introduces undergraduate students to discrete mathematics and the foundations for modern computer science. Beyond learning a set of tools and techniques, a major goal of this course is to train students in how to think logically and mathematically when approaching a problem to solve. Students will learn proof techniques using mathematical logic and see how this informs algorithm design. Students will also learn combinatorial analytical techniques (i.e. counting or enumerating objects) in order to solve computational problems or analyze algorithms. Finally, students will be exposed to discrete data structures: implementations of mathematical structures useful for designing algorithms.

This course will introduce you to a number of mathematical modeling concepts including:

1. Sets
2. Logic
3. Number Theory. Cryptography.
4. Proofs
5. Sequence, Functions
6. Relations. Relational databases.
7. Graph Theory. Trees.
8. Probability
9. Combinatoric
10. Algorithm efficiency

At the end of this course, a successful student will be able to:

- Formulate common language propositions into symbolic logical statements and assess their truth values
- Manipulate, simplify, restate, and relate symbolic logical statements
- Describe and apply different proof techniques such as induction, proof by contradiction,
arguing contrapositive, utilizing the pigeon-hole principle, etc.
- Identify when different proof strategies are applicable to certain problems
- Describe mathematical sets, set operations, and functions and relate these to discrete data structures
- Utilize counting techniques (such as permutations, combinations, binomial coefficients, and their associated identities) in order to solve computational problems
- Describe and apply core concepts in discrete probability and understand how these relate to the analysis of algorithms.

Utilize discrete data structures (like graphs and trees) to express and solve algorithmic and computational problem.

**Modality:** Online asynchronous. All lectures are pre-recorded and are made available on day one. Homework assignments are online (via WebAssign). The two exams (midterm and final, 1 hour 15 minutes each) will be proctored via zoom. You are supposed to have a webcam with a wide view capturing you and the entire desk.

Required Course Materials:

*DISCRETE MATHEMATICS WITH APPLICATIONS, 5th edition, by Susanna S.*

*Epp*. Your professor will provide you with a link to register at WebAssign and to purchase the 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 course outline is tentative, and it will be modified depending on the pace of the class.

**Week1**

Introduction to Sets Theory.
Propositional Logic and Truth table.
Logical Equivalence, Predicate Logics.
Direct Proof Method.

Week2
Number Theory.
RSA: Encryption and Decryption.
Proof by Contrapositive and Contradiction.
Sequences. Proof by induction.

Week3
Set theory again.
Boolean algebras. Russel paradox.
Cryptographic hush functions.
Relations. Graphs.

Week4
Relations on sets.
Relational databases.
Modular arithmetic and cryptography. PERT and CPM.
Introduction to counting and probability.

Week5

Theory of graphs and trees.
Binary search trees. Minimum spanning trees.
Kruskal's, Prim's and Dijkstra's algorithms.
Algorithm efficiency. Divide and conquer algorithms.

**Detailed Course Outline:**

| Week | Chapter | Topic |
|---|---|---|
| | 2 | The logic of compound statements. Logical form and logical equivalence. Conditional statements. Valid and invalid arguments. |
| **1** | | Application: Number systems.<br>Application: digital logic circuits. |
| | 3 | The logic of quantified statements.<br>Predicates and quantified statements. |

| | | |
|---|---|---|
| | | Statement with multiple quantifiers. Arguments with quantified statements |
| | 4 | 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. |
| **2** | | 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. |
| | 5 | Sequences. Geometric sequence. Principle of mathematical induction. Inductive reasoning. Proving divisibility properties, proving inequalities. 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. |
| | 6 | 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. |
| **3** | | Disproving an alleged set property. The number of subsets of a set. “Algebraic proofs” of set identities. Boolean algebras. Russel’s paradox. |

| | | |
|---|---|---|
| | 7 | 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. |
| | 8 | Relations on sets. The inverse of a relation. Directed graph of a relation. N-ary relations and relational databases. Reflexive, symmetric and transitive properties. The transitive closure. Equivalence relations. Equivalence classes. |
| 4 | | 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. |
| | 9 | 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. |
| | | R-combinations and multisets. Pascal's formula and the binomial theorem. Probability axions. Expected value. Conditional probability, Bayes' formula and independent events. |
| | 10 | Theory of graphs and trees. Trails, paths and circuits. Connectedness. Euler and Hamiltonian circuits. Matrices and directed graphs. Isomorphism of graphs. Invariants. |
| **5** | | 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. |
| | 11 | 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/expectations**: The main course material will be presented through lectures. Students are advised to keep pace with the course material as it is being presented. Consequently, students should endeavor to attend class and discussion sessions, and spend sufficient time working on assigned homework problems. If for any reason a student misses a class, he/she should endeavor to obtain the notes and learn the missed material. Students should not hesitate to ask questions or seek additional assistance to ensure that they are staying on pace with the class.

**Examinations**: There will be a midterm exam plus one cumulative final exam. The exams will contain problems to solve and definitions, brief explanations of concepts, and simple proof.