Syllabus of Record

Program: CET Siena
Course Code / Title: (SN/CS 3120) Discrete Mathematics and Theory 2
Total Hours: 45
Recommended Credits: 3
Primary Discipline / Suggested Cross Listings: Computer Science / Data Science
Language of Instruction: English
Prerequisites / Requirements: For UVA students: CS 2100 Data Structures and Algorithms 1 or CS 3100 Data Structures and Algorithms 2. For students from another institution: previous course work in data structures at the same level.

Description

The goal of this course is to understand the fundamental limits on what can be efficiently computed in our universe and other possible (or imaginary) universes. These limits reveal deep and mysterious properties about information, communication, and computing, as well as practical issues about how to solve problems.

Two fundamental questions about any problem are:

1. Can it be solved using a machine of a certain type? (computability)
2. How much does it cost to solve it? (complexity)

We explore these questions by developing abstract models of computing machines and reasoning about what they can and cannot compute efficiently. We also look at some applications in cryptography that take advantage of problems being hard to solve, and what can be done when a problem cannot be solved or is too expensive to solve.

Objectives

At the end of this course students will be able to:

- Improve their mathematical thinking skill and habits, including thinking precisely about definitions, stating assumptions carefully, critically reading arguments, and being able to write convincingly.
- Be able to understand both finite and infinite formal models of computation and to reason about what they can and cannot compute.
- Understand both intuitively and formally what makes some problems too expensive to solve, and what can be done in practice when an unsolvable or intractable problem is encountered.

- Reason formally about the cost of computation, and be able to prove useful bounds on the costs of solving problems, including showing that certain problems are intractable.
- Learn about some interesting aspects of theoretical computer science, including cryptography and machine learning.

Course Requirements
Active participation is essential in this course. Students are expected to attend each class and field-based course component, as outlined in the CET Attendance Policy. Students are expected to read all assigned materials before the relevant class session and come prepared to participate thoughtfully in class discussions. Graded assignments include:

- Problem sets: 10 problem sets that consist of 4-7 problems solved collaboratively
- Quizzes: 6 quizzes that consist of 1-3 problems completed individually

.

Grading
The final grade is determined as follows:
- Participation: 20%
- Problem sets: 35% (each problem set worth 3.5% of total grade)
- Quizzes: 45% (each quiz worth 7.5% of total grade)

Readings
Barak, Boaz. *Introduction to Theoretical Computer Science*. Creative Commons, 2022.
https://introtcs.org/public/index.html

Outline of Course Content
Topic 1: Induction, Representation, Cardinality
- Review discrete math background, in particular proof techniques
- Binary strings
- Using functions to reason about the size of sets

Topic 2: Countability and Uncountability
- Similarities and differences between theory and practice of computing
- Infinities
- Limits of computing

Topic 3: Circuits
- Simple models of computing and complex behavior
- Theoretical models of computing (straight line programs and Boolean circuits) to practical implementations (Python and integrated circuits)

- Powers and limitations of a model of computing
- Compare models of computing

Topic 4: Circuit Complexity, EVAL and Asymptotics
- Implement and use the EVAL function
- Classify functions by complexity
- Asymptotic operators (Big-O, Ω, and Θ)
- Functions in $O(f(n))$, and membership or non-membership

Topic 5: Deterministic Finite State Automata and Regular Expressions
- Unbounded computation
- The connection between functions and languages
- Finite State Automaton and Deterministic Finite Automaton
- The power of DFAs
- How to interpret Regular Expressions, define them formally, and reason about their capabilities

Topic 6: Non-Deterministic Automata, Closure, REGEX=NFA
- Understand the proof that Regular Expressions are equivalent in power to Deterministic Finite Automata
- The power and strangeness of nondeterminism

Topic 7: Limits of NFAs/REGEX, Turing Machines
- What cannot be computed by a finite automaton and why
- Turing Machines as a model of computing, their powers and limitations
- Different variations on Turing Machines and how they can be transformed
- The definition of computability

Topic 8: Universal Turing Machines, Uncomputability, Self-Reject
- Identify functions that cannot be computed by Turing Machines
- Skills of identifying uncomputable problems
- Strategies for showing new problems to be uncomputable
- Universality in the context of Turing Machines

Topic 9: Reductions, Rice’s Theorem
- Proofs by reduction
- Rice’s Theorem to help understand what is uncomputable
- Turing Machine running time and how changing details of Turing machine models can change how efficiently different functions can be implemented