Objectives and Outcomes

Course Objectives

This course introduces students to methods of Discrete (noncontinuous) Mathematics in Computer Science and lays the foundation for subsequent courses in Algorithms, Software Engineering, Computability Theory, and Systems.

Upon completion of 6.042, students will be able to

1. recognize and explain precise mathematical definitions, and to recognize, explain, and synthesize elementary proofs in Discrete Mathematics, especially proofs by induction. They will be skillful at distinguishing rigorously proven conclusions from merely plausible ones.

2. apply methods from Mathematical Logic, Graph Theory, Elementary Number Theory, and Combinatorics to model and solve discrete optimization problems.

3. use these methods to model computational systems, and to verify computational behavior of elementary data structures, algorithms, and state machines.

4. apply principles of discrete probability to model and solve elementary problems of reliability and estimation.

5. work in a small team to accomplish all the objectives above.

Course Outcomes

Students will be able to:

1. evaluate the soundness of elementary mathematical arguments, identify fallacious *reasoning* (not just fallacious conclusions), and synthesize corrections to proofs.

2. synthesize induction hypotheses and simple proofs in discrete mathematics using the terminology of elementary mathematical logic.

3. apply methods of discrete mathematics to model and solve elementary combinatorial optimization problems, in particular

   (a) define and prove basic properties of finite graphs such as connectedness, colorability, acylicity. Use graphs to model data structures and state machines.
   (b) apply graph theory to solve elementary problems of scheduling and connectivity.
   (c) define and prove basic properties of prime numbers and modular arithmetic; calculate GCD's and evaluate arithmetic expressions in modular arithmetic.
   (d) model current value of future income, population growth, and computational resource requirements using series and recurrences, and then derive closed-form solutions.
   (e) count combinations and permutations of outcomes of discrete experiments such as dice rolls.

4. apply principles of logic and discrete mathematics to analyze, verify, and synthesize elementary algorithms and state machines, in particular

   (a) apply the method of invariants and well-founded ordering to prove correctness and termination of algorithms and state machines.
   (b) use asymptotic notation to describe time and space complexity of algorithms.

5. apply principles of discrete probability to model and solve elementary problems of reliability and risk, in particular

   (a) calculate probabilities and conditional probabilities of combinatorial experiments such as dice rolls.
   (b) calculate means and variances for simple combinatorial experiments and for binomial, geometric, and similar discrete distributions.
   (c) state and determine applicability of the Weak Law of Large Numbers and similar estimates of deviation from the mean to problems of estimation and error tolerance.

6. discuss and solve problems with fellow students in a small team, and prepare and orally present solutions to short problems.