CS 5002: Discrete and Data Structures Fall 2019

Northeastern University — Silicon Valley

General Information

This course introduces the mathematical structures and methods that form the foundation of computer science. We will study basic logic, proof techniques, functions, number theory, algorithms, graphs and trees. We will discuss counting techniques and apply the techniques to estimate the size of sets, the growth of functions, and the time complexity of algorithms. We will examine inductive and recursive definitions of structures and functions, as well as data structures such as arrays, lists, maps, stacks, and queues.

Learning Objectives

At the end of CS 5002, a student should be able to do the following:

1. Convert between decimal, binary, and hex.
2. Understand two's complement representations of negative numbers.
3. Read and write mathematical logical formulas.
4. Use logical operators effectively.
5. Understand the set operators of union, intersection, and complement.
6. Understand relations and functions.
7. Perform basic counting tasks - e.g. counting possible passwords - using permutations and combinations as appropriate.
8. Calculate basic probabilities by counting success outcomes and total outcomes.
9. Apply proof techniques, particularly mathematical induction.
10. Model problems using trees and graphs.
11. Describe the number of operations in an algorithm as a function of the input size, and identify superpolynomial running times as intractable.
12. Understand the meaning of big-O notation.

Required Textbook:

Discrete Math and Its Applications by Kenneth Rosen (8th ed)

You have the following two choices for the format of the textbook:

1. Textbook only
2. Online Learning System "Connect": (10-day free trial / $106)

Course Websites: We will use three websites for this course:

1. **NEU Blackboard** for communicating grades. It contains links to the other two websites.
2. **Piazza** for discussions, communicating with the instructor and teaching assistants. Class slides, homework solutions, quiz solutions, exam reviews, etc. will be posted here.
3. (Optional) **Connect** for electronic textbook, homework assignments, learning activities

Meeting Time: Friday 6:00 PM - 9:15 PM (Local Time — Pacific Time)

Professor: Dr. Sarah H Sellke

Teaching Assistants:

- Rajesh Sakhamuru
- Suyue Jiang
- Tuan Nguyen
- Yibao Hu
- Yuting Chen
- Zach Roney

Office Hours:

Monday	10 AM - 1 PM	Raj
	1 PM - 4 PM	Tuan
	4:15 PM - 6 PM	Zach
Tuesday	9 AM -- Noon	Yuting
	10 AM - 1 PM	Suyue
Wednesday	11:30 AM - 1 PM	Zach
	2 PM - 5 PM	Yibao
	4:15 PM - 6PM	Zach
Friday	2 PM - 5 PM	Sarah
Saturday	10 AM - 1 PM	Suyue / Yibao

Grading and Exams:

Category	Date	Weight
Weekly Assignments	Due Wednesdays	55%
Quizzes	Most Fridays	15%
Midterm Exam	10/18/2019 (6 PM - 9 PM)	15%
Final Exam	12/10/2019 (6 PM - 9 PM)	15%

1. Homework Assignments: will be released no later than Wednesday by 11:59 PM. It is due the following Wednesday by 11:59 PM. Everyone will get one token for late homework. If you use a late homework token, your due date will be extended for a week. If you need a homework extension, please notify me and teaching assistants by a private message on piazza. Additionally, the lowest homework score will be dropped.

Under no circumstances may you share written answers or code (including electronically) or dictate answers verbatim to another student. This will be considered plagiarism, see section **"Academic Integrity"** below.

2. Quizzes on Fridays: Research shows that frequent quizzes is one of the most effective ways of learning. Therefore, we will have a quiz every Friday except on 10/18/2019 and 12/06/2019. Quizzes are closed book quizzes. The lowest quiz score will be dropped. There are no make-up quizzes for any reason.

3. Exams are closed book exams. The midterm exam is scheduled on Friday October 18th, and the final exam is scheduled on Tuesday December 10th, 2019.

Important: You are required to take the final exam during the scheduled time. Please do not make travel plans during the final exam week

Letter grades will only be determined by the following criteria: “A”s going to outstanding work, “B”s going to good but flawed work, “C”s going to problematic work that nevertheless gets the key concepts, “D”s going to work where you’ve learned at least some things, and “F”s going to work that doesn’t show the concepts were learned.

A 94% or above is a guaranteed A, 84% or above is a guaranteed B, 72% or above is a guaranteed C, and 60% or above is a guaranteed D.

Professionalism and Respect:

You are expected to treat your instructor and all other participants in the course with courtesy and respect. Your comments to others should be factual, constructive, and free from harassing statements. You are encouraged to disagree with other students and the instructor, but such disagreements need to be respectful and be based on facts and documentation (rather than prejudices and personalities). Part of the learning process in this course is respectful engagement of ideas with others.

Academic Integrity:

You should be familiar with Northeastern's Academic Integrity policy. I will highlight two ways in which it applies here:
(1) **You may not search the web for homework solutions;**
(2) You may not **copy** solutions to homework problems in any case.

You will generally know a homework solution, as opposed to general web research, when you see it. This includes homework solutions posted for other discrete mathematics classes, students at other universities posting their work, or using any kind of web tool that automatically solves a problem you were meant to do by hand.

Copied solutions, from the web or other students, may result in a zero on the corresponding assignment, or harsher penalties if the problem is more widespread in your work. Students who share their solutions will face similar penalties.

Attempts to violate the spirit of the rules listed here with loopholes will generally be received unkindly and treated as integrity violations anyway.

.

Course Schedule:

Date	Topics
09/06	Overview and Logic
09/13	Number Theory
09/20	Application of Number Theory
09/27	Sets and Functions
10/04	Sequences and Summations
10/11	Review for Midterm Exam
10/18	Midterm Exam
10/25	Counting (Chapter
11/01	Induction and Advanced Counting
11/08	Recursion
11/15	Discrete Probability
11/22	Graphs and Trees
12/06	Review for Final Exam
12/10	Final Exam

Course Topics and Learning Outcomes:

Course Topics	Learning Outcomes
Binary/Hexadecimal representation of numbers 1. Numeric data representation and number bases 2. Fixed- and floating-point systems 3. Signed and two's-complement representations	1. *Explain the reasons for using alternative formats to represent numerical data. [Familiarity]* 2. *Explain how fixed-length number representations affect accuracy and precision. [Familiarity]* 3. *Describe how negative integers are stored in sign-magnitude and two's-complement representations. [Familiarity]* 4. *Convert numerical data from one format to another. [Usage]* 5. Representation of non-numeric data (character codes, graphical data) 6. *Describe the internal representation of non-numeric data, such as characters and strings. [Familiarity]*
Basic Logic 1. Propositional logic 2. Logical connectives 3. Truth tables 4. Predicate logic (primarily so students are familiar with concepts and notation of "for all" and "there exists")	1. *Convert logical statements from informal language to propositional expressions. [Usage]* 2. *Describe how symbolic logic can be used to model real-life situations or applications, including those arising in computing contexts such as circuit design. [Familiarity]* 3. *Apply formal logic proofs and/or informal, but rigorous, logical reasoning to real problems, such as predicting the behavior of software or solving problems such as puzzles. [Usage]*
Sets 1. Venn diagrams 2. Union, intersection, complement Cartesian product 3. Power sets 4. Cardinality of finite sets **Relations** 1. Reflexivity, symmetry, transitivity 2. Equivalence relations, partial orders **Functions** 1. Surjections, injections, bijections Inverses 2. Composition	1. *Explain with examples the basic terminology of functions, relations, and sets. [Familiarity]* 2. *Perform the operations associated with sets, functions, and relations. [Usage]* 3. *Relate practical examples to the appropriate set, function, or relation model, and interpret the associated operations and terminology in context. [Assessment]*

Course Topics	Learning Outcomes
Counting Arguments 1. Set cardinality and counting 2. Sum and product rule 3. Arithmetic and geometric progressions **Solving Recurrence Relations**	1. Apply counting arguments, including sum and product rules, inclusion-exclusion principle and arithmetic/geometric progressions. [Usage] 2. Solve a variety of basic recurrence relations. [Usage] 3. Analyze a problem to determine underlying recurrence relations. [Usage]
Permutations and Combinations	*1. Compute permutations and combinations of a set, and interpret the meaning in the context of the particular application. [Usage]* *2. Map real-world applications to appropriate counting formalisms, such as determining the number of ways to arrange people around a table, subject to constraints on the seating arrangement, or the number of ways to determine certain hands in cards (e.g., a full house). [Usage]*
The Pigeonhole Principle	*Apply the pigeonhole principle in the context of a formal proof. [Usage]*
Discrete Probability 1. Finite probability space, events 2. Axioms of probability and probability measures Conditional probability, Bayes' theorem Independence 3. Integer random variables (Bernoulli, binomial) expectation, including linearity of expectation	*1. Calculate probabilities of events and expectations of random variables for elementary problems such as games of chance. [Usage]* *2. Differentiate between dependent and independent events. [Usage]* *3. Compute a probability using that distribution. [Usage]* *4. Apply Bayes theorem to determine conditional probabilities in a problem. [Usage]*
Basic modular arithmetic	*Perform computations involving modular arithmetic. [Usage]*
Notion of contrapositive 1. Proof by contradiction 2. Disproving by counterexample 3. Induction over natural numbers	*1. Identify the proof technique used in a given proof. [Familiarity]* *2. Outline the basic structure of each proof technique. [Usage]* *3. Apply each of the proof techniques correctly in the construction of a sound argument. [Usage]*
Structural induction	*State the relationship between ideas of mathematical and/or structural induction to recursion and recursively defined structures. [Familiarity]*

Course Topics	Learning Outcomes
Trees and Graphs (tie with Structural Induction)	1. *Illustrate by example the basic terminology of graph theory, as well as some of the properties and special cases of each type of graph/tree (unrooted, rooted, binary, not binary). [Familiarity]* 2. *Demonstrate different traversal methods for trees and graphs, including pre-, post-, and in-order traversal of trees. [Usage]* 3. *Model a variety of real-world problems in computer science using appropriate forms of graphs and trees, such as representing a network topology or the organization of a hierarchical file system. [Usage]*
Basic Analysis Differences among best and worst-case behaviors of an algorithm Asymptotic analysis of upper complexity bounds Big O notation Complexity classes, such as constant, logarithmic, linear, quadratic, and exponential Time and space trade-offs in algorithms	1. *Explain what is meant by "best" and "worst" case behavior of an algorithm. [Familiarity]* 2. *Determine informally the time and space complexity of simple algorithms. Good examples for this are searching and sorting algorithms. Note that these are also covered in 5001. Faculty may wish to coordinate. Some repetition/reinforcement can work very well,if planned. [Usage]* 3. *State the formal definition of big O. [Familiarity]* 4. *List and contrast standard complexity classes. [Familiarity]* 5. *Give examples that illustrate time-space trade-offs of algorithms. [Familiarity]*