CMSC 302
Introduction to Discrete Structure

Course Overview

A few general slides about the subject matter of this course

What's *this* class about?

What are "discrete structures" anyway?

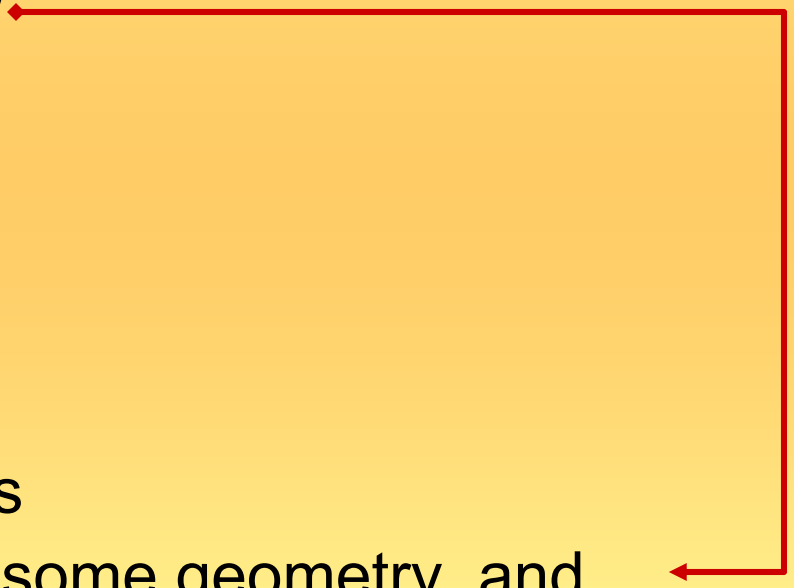
- "***Discrete***" (≠ "discreet"!) - Composed of distinct, separable parts. (**Opposite of *continuous*.**)

  discrete: continuous :: digital:analog

- "***Structures***" - objects built up from simpler objects according to a definite pattern.
- "***Discrete Mathematics***" - The study of discrete, mathematical objects and structures.

Discrete Structures We'll Study

- The Foundations: Logic and Proofs
- Basic Structures: Sets, Functions, Sequences, and Sums (Matrices will come later)
- Relations
- Basics of Counting
- Basics of Probability
- Graphs & Trees
- Induction and Recursion
- Advanced Counting Techniques
- Some algebra, some matrices, some geometry, and some algebra again



Some Notations (We'll Play With)

$\neg p$	$p \wedge q$	$p \oplus q$	$p \to q$	$p \Leftrightarrow q$	$\forall x\, P(x)$
$\exists x\, P(x)$	$\{a_1, \cdots, a_n\}$	$\boldsymbol{Z}, \boldsymbol{N}, \boldsymbol{R}$	$\therefore$	$\{x \mid P(x)\}$	$x \notin S$
$\varnothing$	$S \subseteq T$	$\lvert S \rvert$	$A \cup B$	$\overline{A}$	$\bigcap_{i=1}^{n} A_i$
$f : A \to B$	$f^{-1}(x)$	$f \circ g$	$\lfloor x \rfloor$	$\sum_{\alpha \in S} a_\alpha$	$\prod_{i=1}^{n} a_i$
O, Ω, Θ	$\min, \max$	$a \nmid b$	$\gcd, \operatorname{lcm}$	$\bmod$	$a \equiv b \pmod{m}$
$(a_k \cdots a_0)_b$	$[a_{ij}]$	$\boldsymbol{A}^{\mathrm{T}}$	$\boldsymbol{A} \odot \boldsymbol{B}$	$\boldsymbol{A}^{[n]}$	$\binom{n}{r}$
$C(n; n_1, \cdots, n_m)$	$p(E \mid F)$	R^*	Δ	$[a]_R$	$\deg^+(v)$

Why Study Discrete Math?

- The basis of all of digital information processing: *Discrete manipulations of discrete structures represented in memory.*
- It's the basic language and conceptual foundation of all of computer science.
- Discrete concepts are also widely used throughout math, science, engineering, economics, biology, *etc.*, …
- A generally useful tool for rational thought!

Uses for Discrete Math in Computer Science

- Advanced algorithms & data structures
- Programming language compilers & interpreters.
- Computer networks
- Operating systems
- Computer architecture
- Database management systems
- Cryptography
- Error correction codes
- Graphics & animation algorithms, game engines
- **Just about everything!**

Kinds of Tasks Solved Using Discrete Mathematics

- What are the odds of winning some lottery?
- Are the two computers in network connected?
- How many ways can a password of length 6 be chosen by using alphanumerics only?
- How can I encrypt a message?
- What is the shortest path between two cities?
- Find the quickest path to visit every computer in network, update the software and return to the main computer.
- Find the shortest tour that visits each city only once and then ends in starting city.
- Find the order of targets to shoot at given the denger index of the target and distance?
- How can we prove that there are infinitely many prime numbers?
- How can a list of integers be sorted in decreasing order?
- How many steps are needed to perform such sorting?

Course Outline (as per Rosen, 6th & 7th Ed.)

1. Logic (§§1.1-1.4)
2. Set theory (§§2.1-2.2)
3. Functions (§2.3)
4. Sequences (§2.4)
5. Summations (§2.4)
6. Relations 1 (Ch 8 in 6th and Ch 9 in 7th)
7. Relations 2 (Ch 8 in 6th and Ch 9 in 7th)
8. Basics of Counting (Ch 5 in 6th and Ch 6 in 7th)
9. Basics of Probability (Ch 6 in 6th and Ch 7 in 7th)
10. Graphs & Trees (Ch. 9/10 in 6th and Ch 10/11 in 7th)
11. Advanced Counting Techniques – Recurrences (§7.1 – 7.3 in 6th and (§8.1 – 8.3 in 7th)
12. Algebra & Matrices (partly in §3.8 in 6th and § 2.6 in 7th)

Linear algebra (not in Rosen) - Matrix algebra, & general linear systems, basics of LA only, but with deep understanding what's going on

Topics Not Covered

Other topics we probably won't get to this term:

- Boolean circuits (ch. 11 in 6th and Ch 12 in 7th)
  - You'll learn this in a digital logic course.
- Models of computing (ch. 12 in 6th and Ch 13 in 7th)
  - Most of these are obsolete for practical purposes now anyway
- Abstract algebra (not in Rosen, see Math dept.)
  - Groups, rings, fields, *etc.*

Course Objectives

- Upon completion of this course, the student should be able to:

  - Check the validity of simple logical arguments.
  - Check the correctness of simple algorithms.
  - Creatively construct simple valid logical arguments.
  - Creatively construct simple correct algorithms.
  - Understand basic discrete structures – graphs, trees.
  - Correctly read, write and analyze various types of structures using standard notations.