**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
- Graphs
- Trees
- Induction and Recursion
- Advanced Counting Techniques
- Some algebra, some matrices, some geometry, and some algebra again

---

**Some Notations (We’ll Play With)**

- \( \neg p \)
- \( p \land q \)
- \( p \lor q \)
- \( p \rightarrow q \)
- \( p \iff q \)
- \( \forall x P(x) \)
- \( \exists x P(x) \)
- \( \{a_1, \ldots, a_n\} \)
- \( \mathbb{Z}, \mathbb{N}, \mathbb{R} \)
- \( \text{mod} \)
- \( (x | P(x)) \)
- \( x \not\in S \)
- \( \emptyset \)
- \( S \subseteq T \)
- \( |S| \)
- \( A \cup B \)
- \( \overline{A} \)
- \( \bigcap_{i=1}^{n} A_i \)
- \( f : A \rightarrow B \)
- \( f^{-1}(x) \)
- \( f \circ g \)
- \( \lfloor x \rfloor \)
- \( \sum_{a_i} \)
- \( \prod_{i=1}^{n} a_i \)
- \( a \mid b \)
- \( \text{gcd}, \text{lcm} \)
- \( a \mod b \)
- \( a = b \mod m \)
- \( (a_1, \ldots, a_n) \)
- \( [a]_r \)
- \( A^T \)
- \( A \circ B \)
- \( A^{\ast+} \)
- \( \binom{n}{k} \)
- \( \binom{n}{r} \)
- \( C(n, k) \)
- \( p(E | F) \)
- \( R' \)
- \( \Delta \)
- \( [a]_v \)
- \( \deg^* (v) \)
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!

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. Graphs (Ch. 9 in 6th and Ch 10 in 7th)
9. Trees (Ch. 10 in 6th and Ch 11 in 7th)
10. Induction & Recursion (Ch 4 in 6th and Ch 5 in 7th)
12. Algebra & Matrices (partly in §3.8 in 6th and § 2.6 in 7th)

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.

Think!