University of Florida
Dept. of Computer & Information Science & Engineering

# COT 3100
# Applications of Discrete Structures

Dr. Michael P. Frank

Slides for a Course Based on the Text
*Discrete Mathematics & Its Applications*
(5th Edition)
by Kenneth H. Rosen

# Module #0:
# **Course Overview**

A few general slides about the subject matter of this course.

10 slides, ½ lecture

# What is Mathematics, really?

- It's *not* just about numbers!
- Mathematics is much more than that:

> Mathematics is, most generally, the study of any and all *absolutely certain* truths about any and all *perfectly well-defined* concepts.

- But, the concepts can relate to numbers, symbols, visual patterns, or *anything*!

# So, what's *this* class about?

What are "discrete structures" anyway?

- "***Discrete***" (≠ "discreet"!) - Composed of distinct, seperable 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.

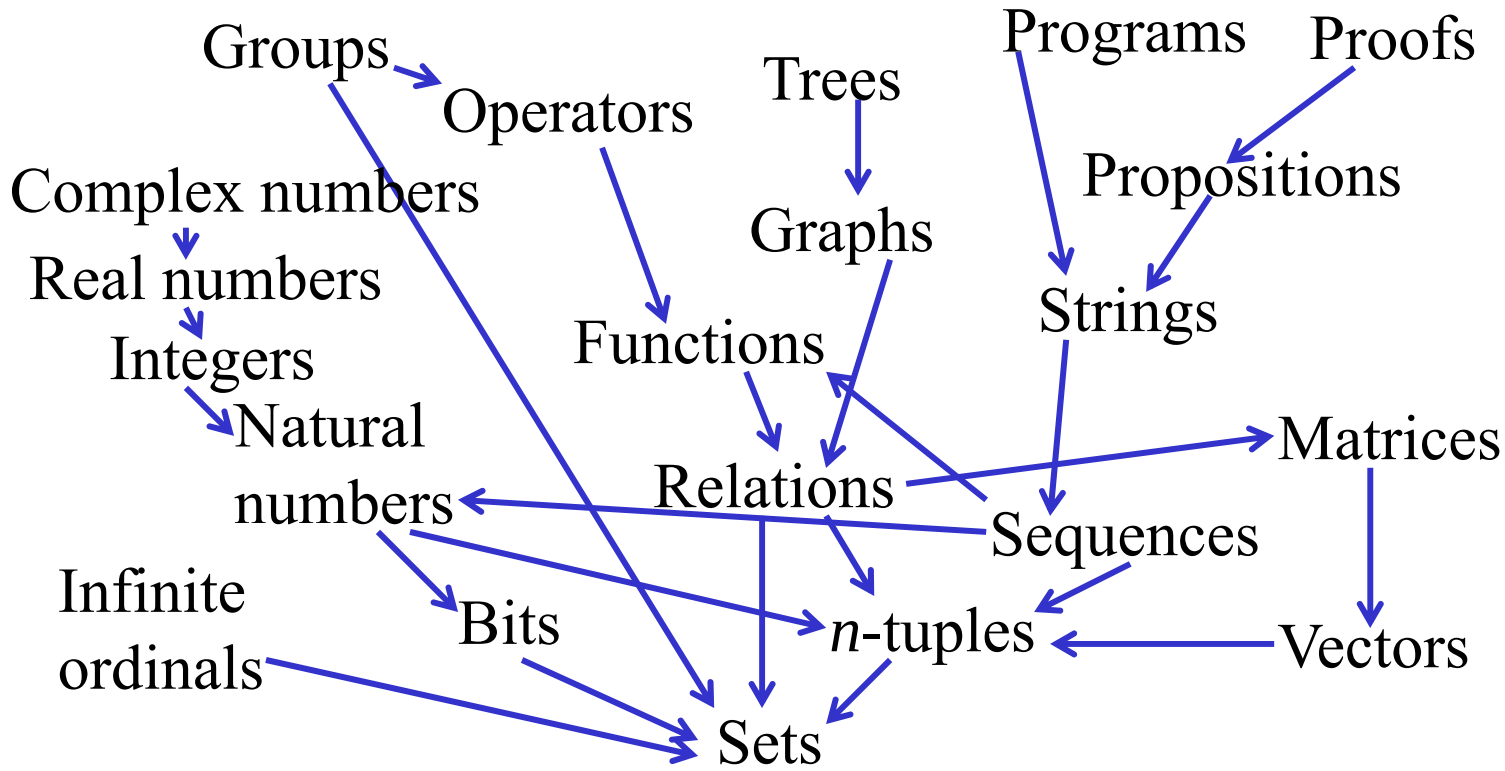
# Discrete Structures We'll Study

- Propositions
- Predicates
- Sets
- (Discrete) Functions
- Orders of Growth
- Algorithms
- Integers

- Proofs
- Summations
- Permutations
- Combinations
- Relations
- Graphs
- Trees

# Relationships Between Structures

- "→" :≡ "Can be defined in terms of"



# Some Notations We'll Learn

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

# 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!

Instructors: customize topic content & order for your own course

# Course Outline (as per Rosen)

1. Logic (§§1.1-1.4)
2. Proof methods (§1.5)
3. Set theory (§§1.6-1.7)
4. Functions (§1.8)
5. Algorithms (§2.1)
6. Orders of Growth (§2.2)
7. Complexity (§2.3)
8. Number Theory (§2.4-2.6)
9. Matrices (§2.7)
10. Proof strategy (§3.1)
11. Sequences (§3.2)
12. Summations (§3.2)
13. Inductive proofs (§3.3)
14. Recursion (§3.4-3.5)
15. Combinatorics (ch. 4)
16. Probability (ch. 5)
17. Recurrences (§6.1-6.3)
18. Relations (ch. 7)
19. Graph Theory (chs. 8+9)

# Topics Not Covered

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

21. Boolean circuits (ch. 10)
    - You'll learn this in a digital logic course.
22. Models of computing (ch. 11)
    - Most of these are obsolete for engineering purposes now anyway
23. Linear algebra (not in Rosen, see Math dept.)
    - Matrix algebra, & general linear algebraic systems
23. 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.
  - Describe the definitions and properties of a variety of specific types discrete structures.
  - Correctly read, write and analyze various types of structures using standard notations.