CISC 1100/1400
Structures of Comp. Sci./Discrete Structures
Chapter 0
Introduction

Arthur G. Werschulz

Fordham University Department of Computer and Information Sciences

Summer, 2017

Welcome to CISC 1100/1400!

- A computer science course, seasoned with a soupçon of math.
- CISC 1100 and 1400: Count towards the mathematical and computational reasoning requirement of the Fordham Core Curriculum.
- CISC 1400: Required course in Computer Science and Information Science majors
- Also used occasionally as an elective.

About your host

- Dr. Arthur G. Werschulz
- Office Hours: MTWR noon–1:00 or by appointment
- Office: LL 815D
- Phone: 212-636-6325
- Email: `agw@dsm.fordham.edu`
- Class email list: `structures-cs@dsm.fordham.edu`

Objective and desired outcomes

- Objective: To develop the necessary abstract reasoning abilities while learning to succeed in a mathematical and computer environment
  CISC 1400: Develop some of the math background needed in later CISC courses
- Desired outcomes:
  - Be able to analyze and understand common math notation
  - Be able to develop solutions to mathematical problems
  - Be able to use a well-defined methodology to reason about math
  - Be able to develop solution to multi-step reasoning problems

Available resources

Textbook — Lyons *et al.*, *Fundamentals of Discrete Structures.* Second Edition, 2012.

Website — `http://www.dsm.fordham.edu/~agw/structures`

Instructor — He would love to help you out. Take advantage of office hours and email!

Things you really must know about

Attendance — Really just short of mandatory. We are all busy people but I need to have you here for all 16 sessions. Unexcused absences or missing more than 4 classes will lower your course grade

Homework — Expect to spend approximately 6 hours each week on work. We'll discuss each day's homework at the next class session. So either know it, or be ready to ask about it!

Grading — As follows:

- Participation: 10%
- Homework: 30%

	CISC 1100	CISC 1400
Written homework	15%	20%
Computer projects	15%	10%

- Midterm exam: 30%
- Final exam: 30%

More things you really must know about

Computer projects — Are designed to be challenging but ultimately doable. Don't give up—I don't expect you to know how to do it from the gun but I expect you to work at them.

Exams — Keep these dates in mind

- **Midterm exam:** Monday, June 19.
- **Final exam:** Thursday, June 29.

Academic integrity — In short: the work you do should be your own. You are only allowed help from authorized sources or when I explicitly permit it. You should read Fordham's academic integrity policy to know all your rights and all the rules

What's discrete mathematics?

- **Continuous mathematics:** deals with objects that can take on a continuous (smooth) set of values (high school algebra, trigonometry, . . .)
- **Discrete mathematics:** deals with objects that can only assume distinct, separated values
  - Sequences, sets
  - Logic
  - Relations, functions
  - Counting, probability
  - Graphs
- Useful for modeling many real-world objects (e.g., the Internet)
- Especially useful for computer problem solving
- Very practical!

We start with sets

- Sets are everywhere ...
  - The group of all students in our class is a set.
  - The group of all juniors in our class is a set.
  - The set of all Facebook members who are not LinkedIn members.
- Some sets are *subsets* of other sets. Example?
- Some sets are *disjoint*—they have no common elements. Example?
- Can do certain *operations* on sets (union, intersection, complement, ...)

Sequences

- People like to see patterns.
- What's the (most "reasonable") next number in each of the following sequences?
  - $2, 4, 6, 8, \dots 10$
  - $1, 2, 4, 8, \dots 16$
  - $1, 1, 2, 3, 5, 8, \dots 13$
  - $1, 2, 6, 24, 120, \dots 720$

Can't predict infinite sequence from finite information! *Any* number could be correct for the next term!

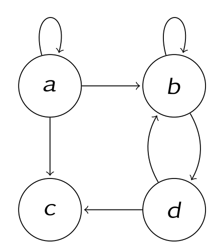
- $2, 4, 6, 8, \dots 15$!!!!
- Why? Let $a_n = \frac{4}{3}n - \frac{1}{2}n^2 + \frac{1}{6}n^3$. Then

n	1	2	3	4	5	6
a_n	1	2	4	8	15	26

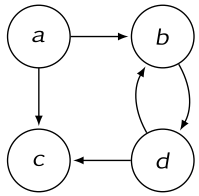
From sets to relations ...

- Among the set of all students at Fordham University, some pairs are special:
  - The pairs in which the students are Facebook friends of each other.
  - The pairs having the same birthday.
  - The pairs taking a specific class during a given term (e.g., CISC 1100/1400 during the first summer term).
  - The pairs that have ever taken a class in common during a given term (e.g., Spring, 2017).
  - The pairs in which the first is older than the second.
- Any such set of pairs is a *binary relation* on the set of Fordham students.
- More generally, can have relations involving *different* sets:
  - Between students and classes: which classes are being taken by a given student?
  - Between people and email addresses: what are a given person's email addresses?
- Relational data bases: needed for e-commerce

Relations may represented by graphs



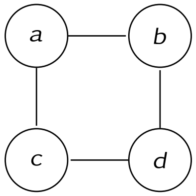
Visualizing relations with directed graphs



This graph could represent:

- Pairs of people, in which the first has sent an email to the second.
- Part of a street map.

Visualizing relations with undirected graphs

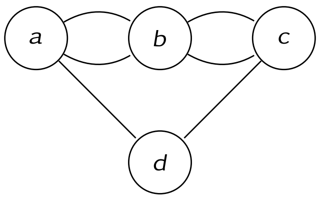


This graph could represent:

- Friendship within Facebook.
- Connections within LinkedIn.

Graph problems

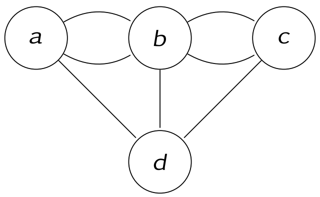
Can you draw the picture



without lifting the pencil or retracing any part of the figure?

Graph problems

Can you draw the picture



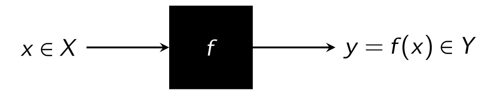
without lifting the pencil or retracing any part of the figure?

Real-world applications using graphs

- Computer networks: how to send data (URL request you type in browser) from your home computer to a web server?
- Facebook: how to suggest new friends?
- Engineering: how to connect five cities to via a highway with minimal cost?
- Scheduling: how to assign classes to classrooms so that minimal number of classrooms are used?

Functions: a special kind of relation between two sets

- ... where each element in the first set is related (mapped) to *exactly one* element in the second set.
- From a certain point of view, describes *any* computation:



- Function composition and decomposition: useful in software design
- Examples of simple functions:
  - "Birth date of" is a function from people to calendar dates (but not vice versa!).
  - "Social security number" is a function from the set of people having SSNs to the set of assigned SSNs (and vice versa).

Our class: birthday remark

- Someone says:

  There are at least two students in the class that were born in the same season.

  Do you agree?
- **Pigeonhole principle:** If you put m pigeons into n pigeonholes, where $m > n$, then there is a pigeonhole containing at least two pigeons.

Another pigeonhole principle example: choosing a pair of socks

- Suppose that you have three different kinds of socks.
- Suppose further that you shut your eyes and reach into your sock drawer.
- How many socks must you choose to guarantee that you'll pick a pair?
- Four.

What is the size of a set?

- Simple questions:
  - What is the *cardinality* (size) of the set? (How many students are in the class?)
  - In how many ways can we represent a class representative?
- Harder questions:
  - In how many ways can we elect a representative and an alternate?
  - In how many ways can we choose . . .
    - a 2-person study group?
    - a 3-person study group?

Probability: How likely is something to happen?

- Suppose I choose one person randomly from the class. How likely are you to be chosen? $\frac{1}{9}$
- Harder questions:
  - Suppose I choose two people randomly from the class.
    - How likely are you to be chosen?
    - How likely are you and your neighbor to be chosen?
  - What's the probability of winning New York State Lotto (pick 6 out of 59)?
  - What about MegaMillions or PowerBall?

Logic: A tool for problem solving

- Your friend tells you:

  If the birds are flying south and the leaves are turning, then it must be fall. Fall brings cold weather. The leaves are turning, but the weather is not cold. Therefore the birds are not flying south.
- Do you agree with her?
- Is her argument valid? sound? (what's the difference)?

Let's analyze her argument

- Suppose the following are true:
  - If the birds are flying south and the leaves are turning, then it must be fall.
  - Fall brings cold weather.
  - The leaves are turning but the weather is not cold.
- Can one conclude "the birds are not flying south"?

Let's analyze her argument (cont'd)

- We'll do a "proof by contradiction".
  - Assume that the birds are flying south.
  - Since (in addition) the leaves are turning, it must be fall.
  - Fall brings cold weather. So it must be cold.
  - But it's actually not cold!!
- Contradiction! So our assumption that the birds are flying south must be wrong. □

Algorithms

- Google's *PageRank* algorithm: finding relevant web pages.
- *Prim's algorithm*: how to connect all of the homes in a town using the least amount of cable.
- RSA *encryption algorithm*: public-key cryptography makes secure e-business possible.
- *Dijkstra's algorithm* for shortest paths between cities.
- α, β *pruning algorithm*: improves the performance of game playing (e.g., chess) programs by quickly eliminating moves that are provably sub-optimal.
- The *Sutherland-Hodgman polygon clipping algorithm*: speeds up the rendering of images for computer graphics and video game programs by removing objects that do not fall into the "camera's" field of view.

Our list of topics:

- Sets
- Sequences
- Logic
- Relations
- Functions
- Counting
- Probability
- Algorithms (maybe, but definite for CISC 1400)
- Graph theory (maybe, but definite for CISC 1400)

Goals for this course

- Master the basics of discrete mathematics
- Develop mathematical and computational reasoning abilities
- Become more comfortable and confident with both mathematics and computation

Discrete mathematics is essential for computer problem solving

- Model real-world entity
  - Student records in a registration system → elements of a set
  - Nodes in a network → vertices in a graph
- Develop/identify an algorithm solving a particular problem
  - Search for a student record (using ID number)
  - Query for a course having a particular prefix (e.g., "CISC").
  - Find shortest path in a graph
- Implement algorithm using a programming language that computers "understand"

Computer projects

- We will learn basic website and web page design
  - Build your own website
  - Learn a bit about HTML, JavaScript, . . .
- CISC 1100: Use the Alice system to build 3D animation clips (cartoons, simple games, . . .)