CSCI 2270 – CS 2: Data Structures

University of Colorado
Boulder

Topics

- A little bit about me…
- Syllabus (info about this course)
- What is a data structure?

A little bit about me…

- Born in SD, moved to CO in 2018
- Began teaching for CU Boulder in 2019
- Worked as Software Engineer, Web Developer, Director of Development and Technology, Tech Lead Software Architect
- https://www.colorado.edu/faculty/ashraf-asa
- My door is always open! I teach because I care about your success!

Where are you coming from?

- Presumably from CSCI1300/CSCI1320
- You should be proficient in a programming language, preferably C++
- Learning a new language is part of computer science
- Resources for learning C++
  - Lecture slides/notes from CSCI1300/1320
  - CU Boulder Data Structures CSCI 2270
  - Bucky's C++ video tutorials

Administrative Details

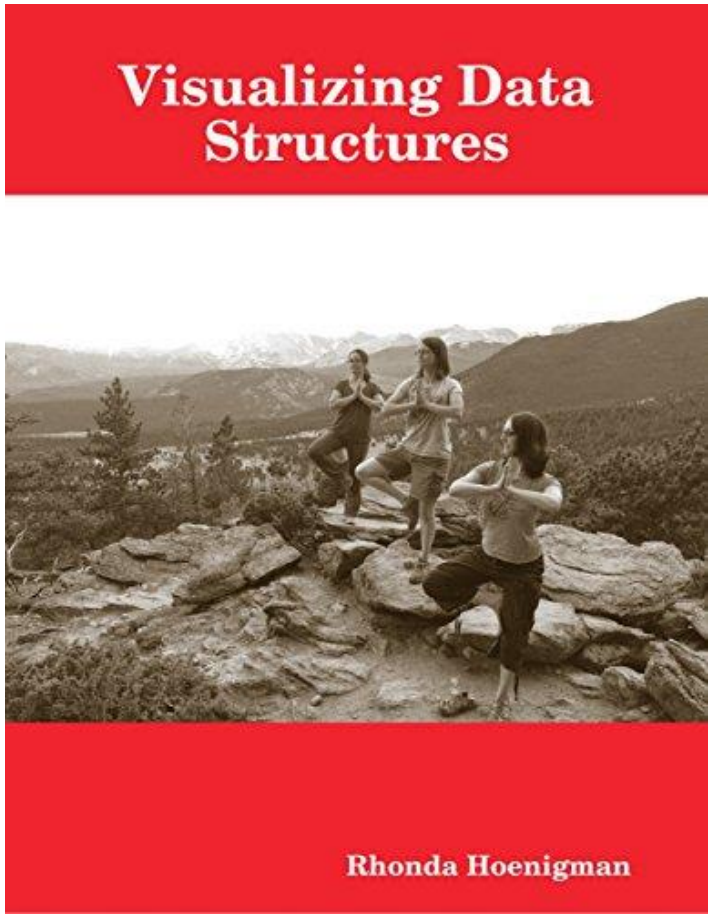
- **Course materials on Computer Science Canvas LMS**
  - Log in with your Identikey and password (colorado.edu)

- **Recitation:**
  - Weekly, 1.25-hour meetings.
  - Recitation activity. Ask questions about assignments and get extra help.

- **Office hours:**
  - Full schedule on Canvas – times and locations

Course Resources

- Required ebook: Visualizing Data Structures, Hoenigman, 2015.
- Available on Moodle for free



Lecture Format

- ~~Bring your laptop to class~~
- Be prepared to take notes/draw diagrams
- Class is theory and implementation
  - Pseudo-code for algorithms
  - Implement algorithms in lecture
- 100% remote/online
- Lectures will be recorded

Piazza

- Online discussion forum where students can ask questions, answer questions, and explore the topics covered in class.
- NOT a complaint board.
- The forum is anonymous to other students, but it is NOT anonymous to the instructors.
- Please be respectful and courteous.
- Students should not depend on getting last minute questions answered here. We cannot require our course staff to work weekends (sometimes they do, but we do not provide a 24hr support service ☺).

Grading Policy

- Recitation 15%
  - Weekly exercises
- Assignments 40%
  - Weekly programming assignments
- Project 15%
  - Given towards the end of the session
- Lecture quizzes 10%
  - 10 questions each week
- Midterms (two midterm exams) 20%
  - Must have a 65% average on the exams to get better than a D+ in the class, regardless of other grades.

Syllabus

- Let's take a look!

- Course description:
  1. Study well-known data abstractions (e.g., stacks, queues, lists, trees) and their representation techniques (e.g., linking, arrays).
  2. Introduces concepts used in algorithm design and analysis including criteria for selecting data structures to fit their applications.

Accommodate

- For those of you who need to submit an accommodation request, Disability Services is excited to introduce Accommodate – a new software tool used to manage accommodation requests and implementation. Accommodate features student and faculty portals to help ease and streamline the accommodation management process for students, staff, and faculty. Check out the Accommodate Portal for more information.

What is Abstraction?

- Focuses on *what* as opposed to *how*
- Separate logical properties from implementation details

What is an Abstract Data Type?

- An Abstract Data Type (ADT) is a collection of data elements and the allowable operations on those elements.
- No implementation details!

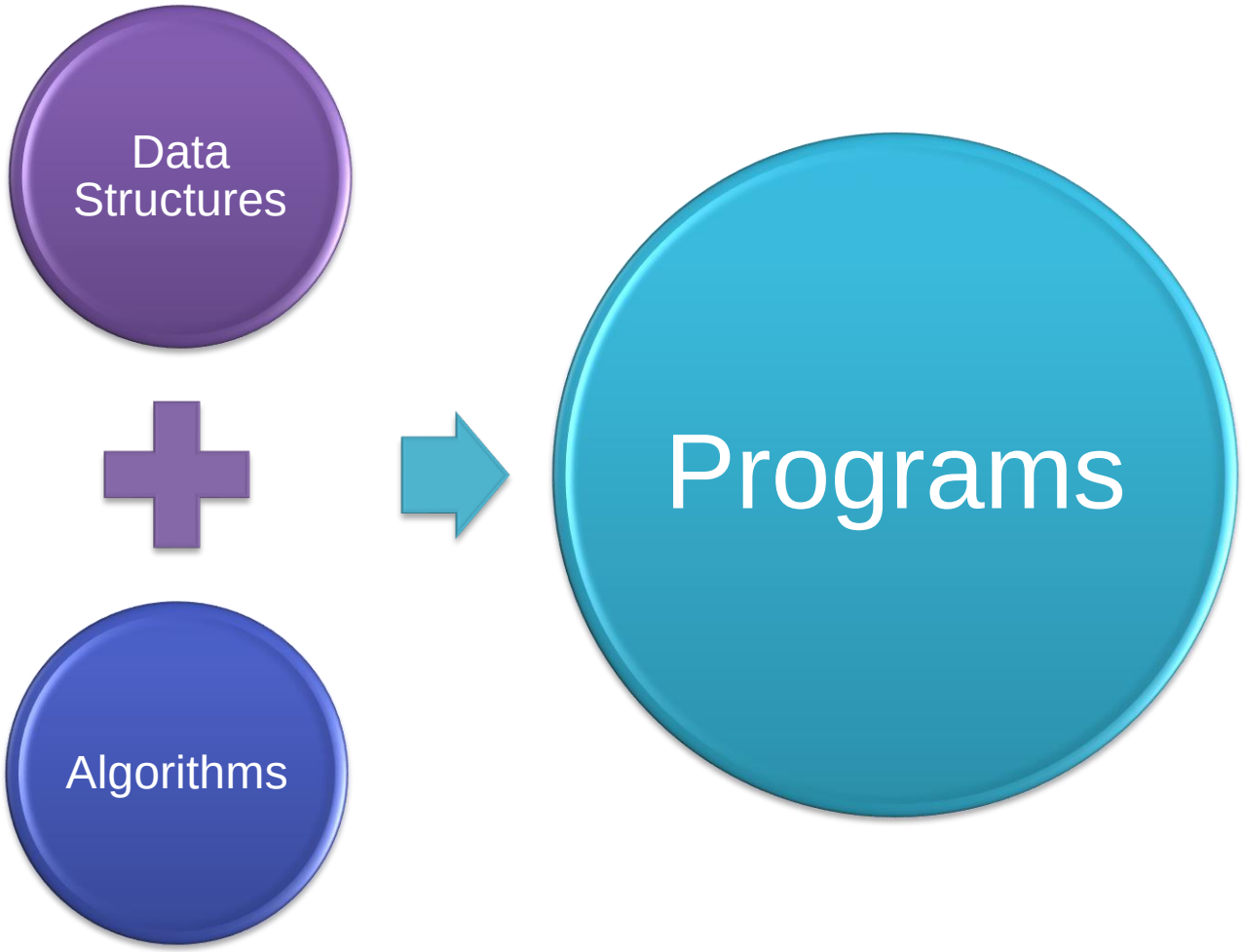
What is a Data Structure?

- Abstract representation of problem data for computational problem-solving
- In layman's terms, a way to store data!

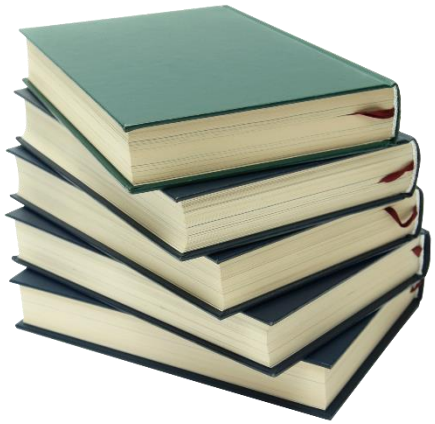
What is an Algorithm?

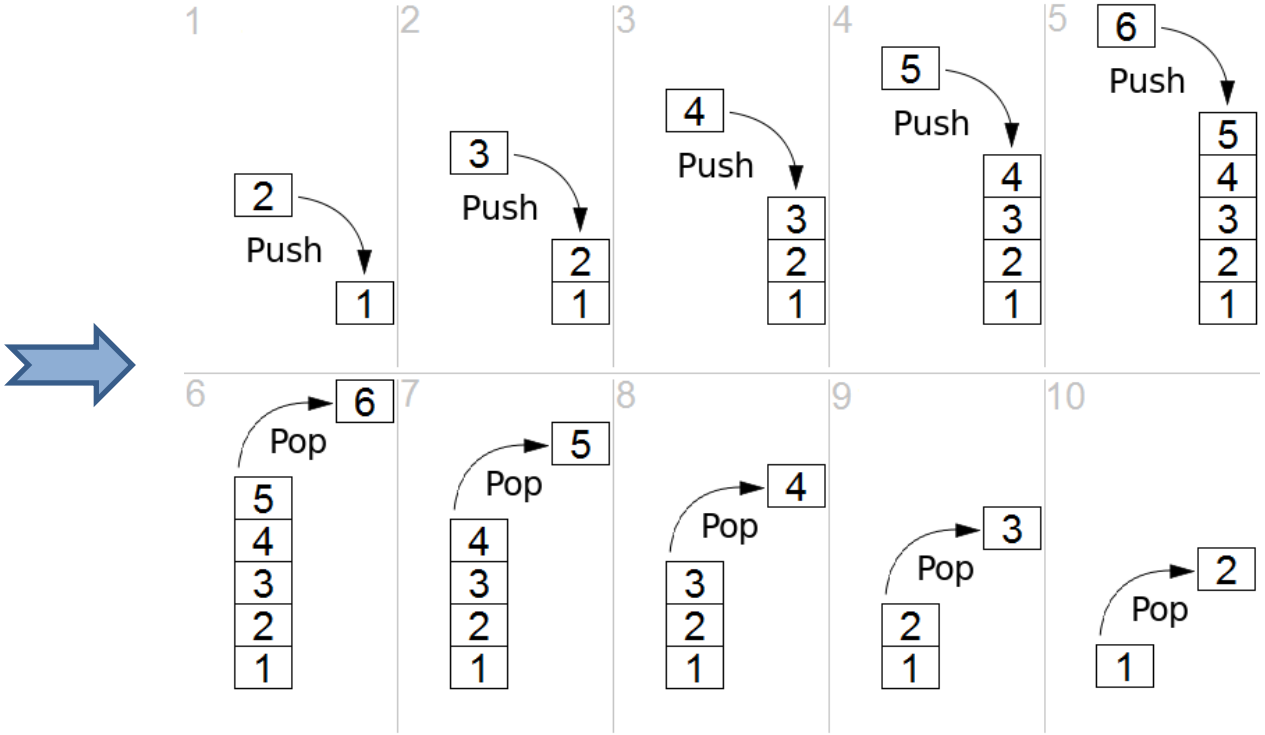
- Abstract recipe to manipulate data for computational problem solving

...which results in...

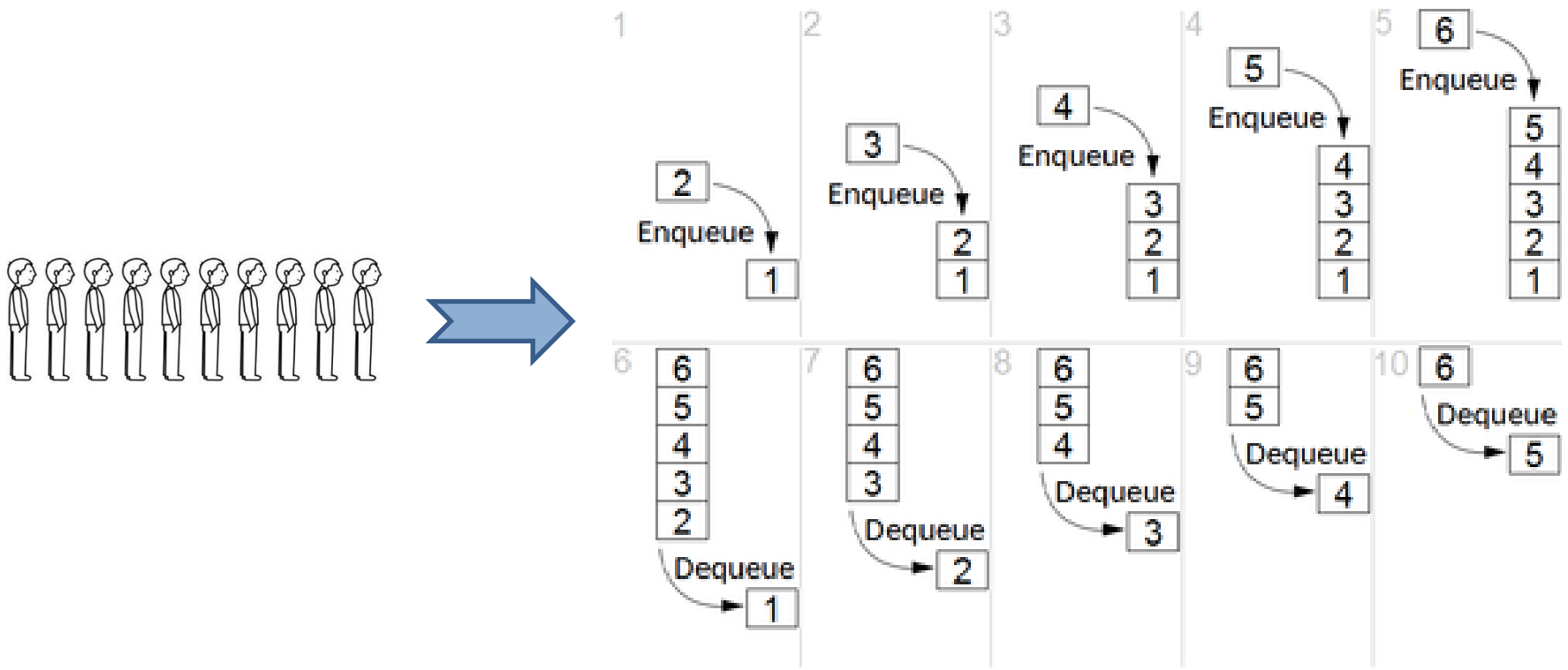


Examples of Data Structures - Stack

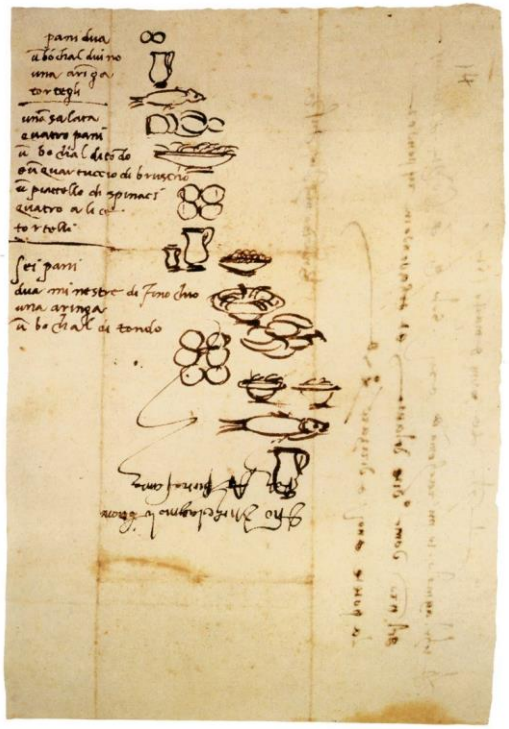


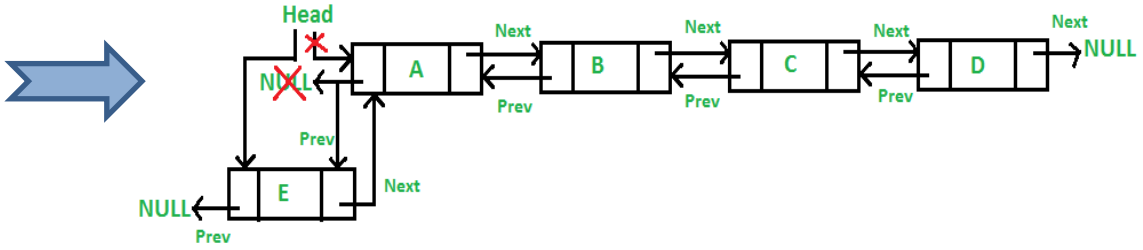


Examples of Data Structures - Queue

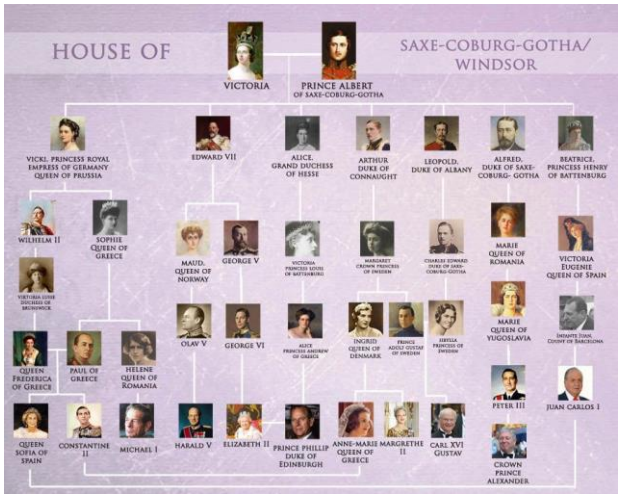


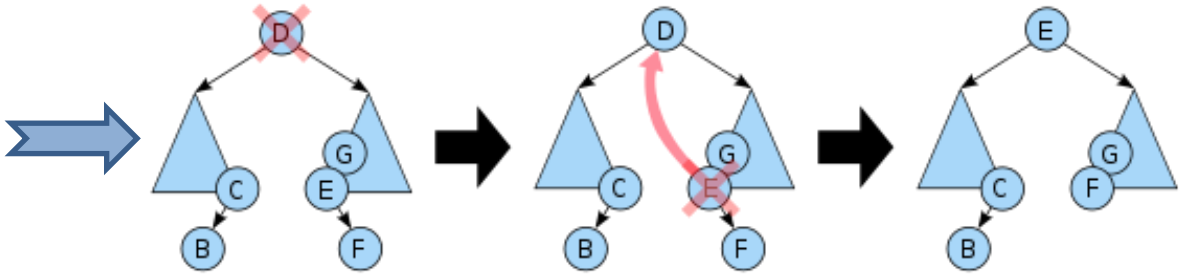
Examples of Data Structures - Lists





Examples of Data Structures - Trees





Asymptotic Analysis

- In reference to the second part of the course description…
- How does an algorithm scale as data becomes really large?

What will you learn in this class?

- How to build data structures
  - E.g.: dynamic arrays, linked lists, stacks, queues, trees, graphs, hash tables
- Why one data structure is better than another for a certain problem.
- Complexity of operations on data structures
  - Search
  - Insert
  - Delete

Have Fun!

Questions?