

Intro: What Is a Data Structure?

EECS 214, Fall 2017

One definition

A scheme for organizing data to use it efficiently

Data structure goals

- Correctness (does what it promises)
- Efficient use of resources:
 - ▶ Time (for operations)
 - ▶ Space (memory)
 - ▶ Power

Example: array set

How long does it take to find an element? How long to add one?

14	2	65	23	26	80	45
----	---	----	----	----	----	----

Example: array set

How long does it take to find an element? How long to add one?

14	2	65	23	26	80	45
----	---	----	----	----	----	----

What if we sort it?

2	14	23	26	45	68	80
---	----	----	----	----	----	----

Characterizing data structures

- Almost always comes with an *algorithm*
 - ▶ (an effective procedure to a class of problems)
- Usually implements an *abstract data type*
 - ▶ (a set of operations with rules about their behavior)

Example abstract data type: stack

- Operations: push, pop, peek
- Implementations:
 - ▶ Linked list: cons, rest, first
 - ▶ Array?

Example abstract data type: set

- Operations: empty?, member?, insert, union, intersect, size
- Implementations:
 - ▶ Linked list
 - ▶ Array
 - ▶ Binary search tree
 - ▶ Hash table

Related things that aren't really data structures

- File/serialization/interchange formats (e.g., JSON, XML)
- Databases (though they often use very fancy data structures)

Concrete data structures

Concrete data structures

- struct
- array
- linked list (single, double, circular)
- ring buffer
- binary search tree
- adjacency list and adjacency matrix
- binary heap
- union-find
- hash table
- Bloom filter
- dynamic array
- AVL and red-black trees

Other concepts

- Abstract data types
- Asymptotic analysis (big-O notation)
 - ▶ Worst case
 - ▶ Average case
 - ▶ Amortized worst case
- Hashing

Administrivia

Course staff

Instructor: **Jesse** Tov

- Email: jesse@eecs.northwestern.edu
- Office: Ford 2-215
- Office hours: Gladly by appointment

Course staff

Instructor: **Jesse** Tov

- Email: jesse@eecs.northwestern.edu
- Office: Ford 2-215
- Office hours: Gladly by appointment

Undergrad TAs:

- Daniel Zhu
- Ellie Tyger
- Emma McDonnell
- Jared Schifrien
- Matt Cheung
- Nathan Lindquist
- Sameena Khan
- Scott Renshaw
- Vickie Li

Prerequisites

One of:

- EECS 111 and 211
- EECS 230
- or something equivalent

Course structure

- Lectures will be mostly theoretical
- Homework is programming
- Exams cover both

Grading

- Five programming assignments worth 10% each
- Two in-class exams worth 25% each
- The map from numbers to letter grades is at my discretion

Exams

No final! Two in-class exams:

- 1st: Tuesday, October 24th
- 2nd: Thursday, November 30th

Homework

Five programming assignments:

- Four done with a partner
- Language: DSSL2 (Data Structures Student Language 2)

Graded by automated testing (which can be picky) and TAs (pickier still)

No late work accepted

Resources

In person:

- TAs
- Instructor

Online:

- <http://users.eecs.northwestern.edu/~jesse/course/eecs214/>
- Piazza board

Books (optional):

- Udi Manber, *Introduction to Algorithms: A Creative Approach*.
- CLRS (Corman, Leiserson, Rivest, Stein): algorithms

Stealing

Stealing

- Only turn in code you wrote (or consult instructor)
 - ▶ (but you can share tests in this class)
- Avoid poisoning (seeing something you shouldn't)
- Accessory to the crime is as culpable as the criminal
- (Your responsibility to protect your work)

How to avoid stealing

- Start early
- Don't look at others' homework
- Don't post homework code on Piazza
- If you aren't sure, ask course staff

Why not steal?

- You'll be reported to Dean Burghardt,

Why not steal?

- You'll be reported to Dean Burghardt,
- you'll no longer be welcome in this class, and

Why not steal?

- You'll be reported to Dean Burghardt,
- you'll no longer be welcome in this class, and
- **MOST IMPORTANTLY**, you won't learn.

Next time: Boxes and arrows