Overview

- Introduction of data structures and algorithms
- Information about course
**What is an algorithm?**

- Well-defined computational procedure
- Takes set of values as **input** and produces set of values as **output**
- Tool to solve well defined computational problem

---

**Algorithm example: Sorting numbers**

- **Input:** A sequence of $n$ numbers: $\langle a_1, a_2, ..., a_n \rangle$
- **Output:** A permutation (reordering) $\langle a'_1, a'_2, ..., a'_n \rangle$ of input sequence such that $a'_1 \leq a'_2 \leq ... \leq a'_n$
- **Input:** $\langle 31, 41, 59, 26, 41, 58 \rangle$
  **Output:** $\langle 26, 31, 41, 41, 58, 59 \rangle$
- Since sorting is essential, a large number of sorting algorithms exist
A Data-centric View of the World

• Data are important in many disciplines
  • DNA sequences, images, network traffic, sensor readings, maps, customer profiles, etc.
  • COVID-19 related data across all disciplines

• Past ECE focus: processing
  • Computation at core: processors, computers, etc.

• New ECE focus: data
  • Data at core: data centers, data security, data science, etc.

• Need structured approach to handling data

What kind of problems are solved by algorithms?

• Human Genome Project
• Internet: Routing, searches, security
• Electronic commerce
• Commercial enterprises
• ...

Human Genome Project

- Complete mapping and understanding of all the genes of human beings
- ~20,500 human genes
- ~3 billion chemical base pairs that make up human DNA
- Requires algorithms to store and analyze

Infectious Disease Modeling

- Flu forecast
- COVID-19: forecast of death and hospitalization
- Generate computer models (in form of algorithms) to make predictions
- Statistical modeling
Internet

- Routing algorithms that find shortest path and adapt to changes
- Search engines that find pages with particular information
- Encrypted communication
- Queue management

Electronic Commerce

- Ad placements
- Google search
- Recommendations: Netflix, Amazon, etc.
- Hashing for authentication
- Electronic trading
Manufacturing and Commercial Enterprises

• Allocation of scarce resources:
  • Assigning crews to flights
  • Routes for package deliveries
  • Storage and distribution of goods

Why Data Structures?

• Way to store and organize data
• Facilitate access and modification of data
• Different data structures have strengths and weaknesses
• Better suited for a specific algorithm than others
Data Structures and Algorithms

• Data structures
  • Representation and organization of data
• Algorithms
  • Methods for implementing operations on data structures
• Data structures and algorithms are closely related
  • Data structures are often tuned for certain algorithms

Big Data

• Extremely large volumes of data
• Can be computationally analyzed
  • Internet of Things
  • Social media content
• Requires algorithms and data structures to analyze
Artificial Intelligence

• Often “trained” on big data sets
  • Image recognition
  • Voice recognition
• Relies on a large set of algorithms, varying in complexity
  • Linear regression
  • Statistics, probability

Complexity Problems

• Data sets can be very large
• Operations may take a lot of time or memory
  • Efficient implementations can make a big difference
• Examples of large data sets and sizes?
• **Example: Million Song Dataset**

    • Data set of songs: title, artist, recording years, etc.

  • How would you figure out
    • Which artist has recorded most songs?
    • Which song has been covered the most times?
    • What are the most common words in a title?

  • What are potential problems?

---

**What will you learn in the course?**

• How to think about data and operations on data
• How to design data structures for efficient use
• How to determine the efficiency of algorithms
• A set of common data structures and algorithms for typical operations on data (e.g., searching, sorting, etc.)
• Advanced programming techniques (e.g., recursion, etc.)
• How to implement a larger software project with a professional IDE
Python

• We will use Python as programming language
  • Data structures and algorithms are independent of programming language
• This is not a programming course
  • We focus on higher level issues
  • You should already have experience with a programming language (Python, Java, C/C++)
• IDE used in this course: PyCharm
  • Professional environment

PyCharm

• IDE for Python
• https://www.jetbrains.com/pycharm/
• Very important to become familiar with tool!!
• See intro slides for PyCharm (more during this week’s discussion)
Course Organization

• Course web site:
  http://www.ecs.umass.edu/ece241/

Course Events

• Events to attend
  • Lectures
  • Discussions

• Assignments
  • Homework
  • Projects

• Exams
  • 2 exams during semester
  • 1 final exam
**Course Policy**

- Read the syllabus in its entirety!
  - Understand expectations for this course
- Deadlines are posted on web site
  - No extensions for homework or projects
- Assignments and grades managed through Moodle and Gradescope
- No sharing of code
  - Automated checking of submissions

**Office Hours**

- TAs:
  - Tuesdays: 5 PM (Guoyi)
  - Wednesdays 7 PM (Zibin)
- Prof. Zink:
  - Fridays 8 – 9 AM
Properties of Data Structures and Algorithms

Quality Measures for Data Structures

- What do you expect from a good data structure?
Quality Metrics for Algorithms

• What do you expect from a “good” algorithm?

Correctness

• In some cases absolute
  • E.g., what is the sum of all values in array?

• Some algorithms have multiple solutions
  • Examples?

• Some algorithms may have solutions where correctness is hard to define
  • Examples?
Multiple solutions

• Data set for number of wins by pitcher (2011):
  • \{(Halladay, 21),
    (Jiminez, 19),
    (Lester, 19),
    (Price, 19),
    (Sabathia, 21),
    (Wainwright, 20)\}

• Which pitcher has most wins?

Unclear “correctness”

• Make image square:
Running time performance

• “The algorithm finishes in 12 seconds.”
  • How to express performance more accurately?

Memory use performance

• “The data structure requires 727 bytes.”
  • How to express memory requirements in a more meaningful way?
It all depends...

• Quality of data structure depends on operation
  • Example: unsorted “pile” of papers
    • Very fast for adding item
    • Not very fast for finding item
  • Example: papers sorted by last name
    • Slower for adding item
    • Faster for finding item (if search is by last name)
    • However: also slow if search is by first name

Data types

• What are typical data types to store information?
Data types

• Common types
  • Numeric types: integer, floating point
  • Boolean
  • Text types: character, string
  • Composite types
  • References ("pointer")

Algorithms and data types

• Many algorithms can be used with different data types
  • Sorting of integers
  • Sorting of strings
  • Sorting of composite types

• May require small adaptation in code
  • Different comparison functions
  • Different code for “moving data around”

• Why is this point important?
  • I do not want you to leave this course saying: “Useless. We only learned how to sort numbers!”
Objects

- For algorithm discussion, we can think of “objects” in a very general way
- Operations on objects
  - Creation (and destruction)
  - Comparison
- Operations on objects in data structures
  - Insertion
  - Deletion
  - Algorithm-specific operations: search, sort, etc.

Next Steps

- Next lecture “Asymptotic Notation and Merge Sort” on Tuesday
- First discussion TODAY!!
- Will post first HW later today