Preliminaries

CS 331: Data Structures and Algorithms
Michael Lee <lee@iit.edu>
Michael Lee

- lee@iit.edu
- http://moss.cs.iit.edu
- Office: SB 226C
- Hours: Wed/Fri 12:30-1:30
Agenda

- Course overview & Administrivia
  - Prerequisites
  - Topics & Resources
  - Grading
  - Dev environment & Class procedures
Data Structures

- How do we store, organize, and retrieve data on a computer?

& Algorithms

- How can we efficiently (in space/time) carry out some typical data processing operations?
- How do we analyze and describe their performance?
Prerequisites

- I assume you are …
  - fluent in some programming language
  - familiar with procedural & OO paradigms
  - comfortable with development processes:
    - compilation, debugging, testing
Python

- We’ll use the Python programming language to explore data structures & algorithms
- Easy-to-learn, clean (“one obvious way to do” things), and popular language
- Ton of useful, powerful libraries
Topics

- Python crash course
- Algorithmic analysis
- Linear data structures (Lists, Stacks, Queues)
- Hashing and Hashtables (aka Maps)
- Recursion and Trees
Class Resources

1. Course website: moss.cs.iit.edu/cs331
   - static information
   - lecture calendar, slides, external resources, etc.
Class Resources

2. Learning platform: Mimir
   - lab notebooks with built-in tests
   - quizzes/self-assessments
Class Resources

3. Google Colaboratory
   - interactive lecture notebooks
Class Resources

4. Blackboard
   - Final gradebook
Class Resources

5. Discord: discussion forum
   - text/voice chat + screen share
Supplements

- The Python Tutorial (docs.python.org/3/)
- Problem Solving with Algorithms and Data Structures Using Python
Grading

- 35% Machine Problems
- 5% Quizzes / Self-evaluation
- 60% Exams (3 total: 2 midterms + final)
On Exams

- Will be announced at least 2 weeks in advance
- Nominally cumulative (but will focus on material since preceding exam)
- Format/Modality to be determined
Quizzes

- Released periodically on Mimir
- For self-evaluation and exam preparation
- Retake as many times as you wish (but scores released after due date)
Machine Problems

- New programming assignment most weeks
- All assignments are retrieved and submitted on Mimir
  - Provided codebase typically covered in preceding lectures
- Late policy: 7 total late days, distributed as you like throughout semester — once out, no late work accepted!
Jupyter Notebook

- In-browser Python development platform
  - “Cells” can contain plain text, code, output (and more)
  - All lecture notes, demos, and assignments will be distributed as notebook files
Jupyter Notebook

- You can optionally install a notebook server on your own computer for convenience

- See [http://jupyter.org/install.html](http://jupyter.org/install.html) — either JupyterLab & “Classic” Jupyter Notebook are fine (with Python3)
Interactive Lectures

- Lecture notebooks available in course repository
- Open on Google Colab or local Jupyter installation
  - Class is usually one long interactive demo.
- Completed notebooks posted on the class website
§ Demo