Preliminaries

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

- lee@iit.edu
- http://moss.cs.iit.edu
- Office hours by appointment (Tue/Thu 11AM-1PM)
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)
- Mapping structures (Hashtables and Trees)
- Recursion
Class Resources

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

2. Google Colaboratory
   - interactive lecture notebooks
3. Blackboard
   - Final gradebook
Class Resources

4. GitHub
   - Assignment distribution & submission
Supplements

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

- 40% Programming Assignments
- 60% Exams (3 total: 2 midterms + final)
Programming Assignments

- ~10 assignments
- All assignments are retrieved and submitted via GitHub
  - Provided codebase typically covered in preceding lectures
- Late policy: elastic over summer
  - Hard due date for everything: June 24
On Exams

- Tentative dates are on course website (June 3, June 10)
- All exams are cumulative (but will focus on recent material)
- Will be administered online, to be taken on your own computer (or in lab)
- Later exam scores, if higher, will improve earlier ones
scores = [60, 80, 75]
[max(scores[i:]) for i in range(3)]
[80, 80, 75]

scores = [75, 80, 100]
[max(scores[i:]) for i in range(3)]
[100, 100, 100]
Jupyter Notebook

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

- We strongly recommend installing Microsoft Visual Studio Code (VSCode) as an IDE for both lectures & labs

- Install the Jupyter and Python extensions
Interactive Lectures

- Lecture notebooks available in course repository
  - Open on Google Colab or VSCode
    - Class is usually one long interactive demo
- Starter and completed lecture notebooks are available in the course GitHub repository