Preliminaries

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

- lee@iit.edu
- http://moss.cs.iit.edu
- Office: SB 226C
- Hours: Wed/Fri 3:15PM-5:15PM
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
Online resources

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

2. Learning platform: Mimir
   - interactive lab and lecture notebooks (using Project Jupyter)
   - quizzes for self-evaluation
Online resources

3. Blackboard
   - only for grade reporting!
Online resources

4. Piazza: discussion forum
   - all class-related questions
   - monitored by TAs
   - scales *way* better than e-mail
Teaching Assistants (SB 108)

- Section 02: Matthew Anderson
  - Hours: TBA
- Section CPS: Truong Pham
  - Hours: TBA
- Section CPS: Vincent Tran
  - Hours: TBA
- Section CPS: Mohit Jha
  - Hours: TBA
- Section CPS: Ismael Lopez
  - Hours: TBA
- Section CPS: Aakef Waris
  - Hours: TBA
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

- Tentative midterm exam dates published on class website
- Feb 21, Apr 1: coverage will be announced
Machine Problems

- New programming assignment most weeks
- All assignments are retrieved and submitted on Mimir
  - Provided codebase typically covered in preceding lectures
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

- Install via Anaconda (“classic” Jupyter Notebook with Python3) — see http://jupyter.org/install.html
Interactive Lectures

- Lecture notebooks released as 0-point “assignments”
- Open on Mimir (or download into local notebook server) to edit and follow along during class
- Class is usually one long interactive demo. Bring your laptop to follow along!
- Completed notebooks will be posted on the class website
§ Demo