INTRODUCTION TO CS16

CS16: Introduction to Algorithms and Data Structures
Tu/Th 10:30-11:50
Salomon DECI
Seny Kamara
What is 16 About?

Algorithms

“a series of computer instructions to perform a given task”
Why Study Algorithms?

• Core to Computer Science
  • Algs:CS is like Math:STEM

• Extremely useful in practice
  • Age of the universe vs. seconds

• Interesting and elegant ideas
Computer Science is Diverse

- Algorithms
- OS
- Networking
- Soft. Eng.
- Graphics & Vision
- AI
- Crypto & Security
- PL
Web Search in the 90’s

Ask Jeeves
altavista
LYCOS
Yahoo!
Google
What are we learning about?

- **Analysis of algorithms**: big-O, worst-case analysis, amortized analysis, expected running time
- **Design paradigms**: dynamic programming, divide and conquer, greedy algorithms
- **Recursive algorithms**: recurrence relations, induction
- **Elementary data structures**: stacks, queues, trees, hash tables, binary search trees, heaps, graphs
- **Sorting algorithms**: insertion sort, selection sort, heap sort, merge sort, quicksort, radix sort
- **Geometric algorithms**: convex hull
- **Graph algorithms**: depth-first search, breadth-first search, shortest path, minimum spanning tree, topological sort
- **Advanced topics**: mapreduce, Bitcoin, PageRank, numerical algorithms
Welcome to CS16!
Outline

1) Meet your TAs
2) CS 16 Infomercial
3) Analyzing our First Algorithm: SeamCarving
Meet your TAs!
Goals
• Learn fundamental algorithms and data structures
• Find and design new ones
• Reason about them
• Use them
• Prepare you for more CS

Course Work
• Lectures
• 10 Homeworks (30%)
• 4 Projects (30%)
• 2 Exams (25%)
• Sections (10%) req’d!
• In-class worksheets (5%)
• Keep up with website / piazza!
• Reading: Dasgupta (Optional)
Lectures

- Cover various algorithms & data structures
  - How they work
  - Why they work
  - Analyze them
- Activities & discussions
- You are responsible for content in lecture (whether on slides or not)
Textbook

• No required textbook
• Helpful
  • Das Gupta, Papadimitriou and Vazirani
  • Goodrich and Tamassia
Course Page

• Slides
• Notes
• Announcements
• Helpful readings
Piazza

• Announcements
• Questions and answers
• Links to helpful material (blogs, Youtube videos)
Office Hours

- TA office hours are very helpful
  - Try to get unstuck on your own first
- My office hours by appointment
- Questions about HW or projects:
  - Post on Piazza
  - Ask in Section
  - TA office hours
  - Schedule meeting with me
Homeworks (30%)

- 10 HWs
- Due every week
- Python code, proofs, analysis, …
Projects (30%)

- 4 projects in Java
Sections (10%)

• 1 hour/week with 2 TAs
• 6-10 students
• Required!
  • If you miss 3 you fail
  • Lose points for every missed section
• Mini assignments
• Mentor
Exams (25%)

- Midterm (10%)
  - Date: March 23rd, time TBD
- Final (15%)
  - Date: May 16th, 2PM
THUS, FOR ANY NONDETERMINISTIC TURING MACHINE M THAT RUNS IN SOME POLYNOMIAL TIME $p(n)$, WE CAN DEVISE AN ALGORITHM THAT TAKES AN INPUT $\omega$ OF LENGTH $n$ AND PRODUCES $E_{m,\omega}$. THE RUNNING TIME IS $O(p^2(n))$ ON A MULTITAPE DETERMINISTIC TURING MACHINE AND...

WTF, MAN. I JUST WANTED TO LEARN HOW TO PROGRAM VIDEO GAMES.
Collaboration

- Encouraged to collaborate on HWs but
  - Write up HWs by yourself
  - Code by yourself
  - No sharing of code
  - No collaboration on Projects
- You will re-write collaboration policy
- We will use code similarity tester
- Random live audits
  - “what would happen if we did X to your code?”
Collaboration

• You’ll read and rewrite, in your own words, the collaboration policy as part of HW1

• You’re highly encouraged to collaborate on homeworks, but....

• One key point: no code sharing.
  • We’ll use an automated code-similarity tester
    • If you’re clever enough to beat it, you shouldn’t be in this course
  • We may, at random, do a “wire pull test”, i.e., ask you what your code would do if we changed some line to say something different.
Goals

- Learn fundamental algorithms and data structures
- Find and design new ones
- Reason about them
- Use them
- Prepare you for more CS

Course Work

- Lectures
- 10 Homeworks (30%)
- 4 Projects (30%)
- 2 Exams (25%)
- Sections (10%) req’d!
- In-class worksheets (5%)

- Keep up with website / piazza!
- Reading: Dasgupta (Optional)