CS 106B, Lecture 28
Where do we go from here?
Plan for Today

• Where we've been
• Where to go from here
• Your Questions
Where We've Been

• C++ syntax, including streams and strings
  – you're ready to learn other languages now, if you choose

• ADTs
  – How they're implemented, when to use each one

• Recursion and recursive backtracking
  – Algorithmic thinking, more abstract code
  – Some problems require "trying everything" – take CS103!

• Big Oh
  – Analyzing algorithms, understanding tradeoffs

• Pointers and arrays
  – How memory works, manipulating computers at a more basic level
Where We've Been Cont.

- Linked data structures
  - Designing data structures

- Graphs
  - Modeling real-world problems with theory

- Sorting
  - Multiple ways to solve the same problem

- Hashing
  - Multiple uses for one concept
• Major course goals:
  – Handling complexity with abstraction
  – Breaking problems down with recursion
  – Understanding memory and linked data structures
  – Evaluating algorithms and their efficiency
  – Debugging and testing code

• You'll use all these skills going forward
The CS Major

- CS 106A
  - CS 106B
    - CS 107
    - CS 110
    - CS 103
      - CS 109
        - CS 161
• Introduction to CS Theory
  – How to write proofs (great for the logical thinking of programming)
  – How can we model and reason about computers mathematically?
  – Very cool results about what is possible to solve with a computer and what is possible to solve *efficiently* with a computer
• Introduction to CS Systems
  – How does the computer actually store/manipulate memory?
    • C is a powerful language
  – How do the stack and heap work?
  – What is compiled code like?
  – Become a ninja debugger
Where you could go

The CS Major – Tracks

– Artificial Intelligence (CS221 & CS229)
– Biocomputation (CS221 & CS27X)
– Computer Engineering (EE108)
– Graphics (CS148)
– Human-Computer Interaction (CS147 & CS247)
– Information (CS124 & CS145)
– Systems (CS140)
– Theory (CS154, CS16X)
– Unspecialized
– Design your own!
Where you could go

Reasons to become a CS major
- Enjoyment of the courses/material
- Potential for impact
- Camaraderie with other CS majors
- Financial security

Bad reasons to not become a CS major
- Thinking other people are better
- Not knowing what to do with it yet/which track to choose
- Fear of being "stuck" as a software engineer

Someone in the CS major is a jerk (I hope this never happens to you, and it's a personal decision, but we're here to support you)
Where you could go

The CS Minor

- CS 106A
  - CS 106B
    - CS 107
    - CS 103
      - CS 109
Where you could go

The CS Coterm

• The core plus specialization
• Strive for as good of a GPA as possible and to take CS courses to have a track record
• Open to *all* majors
• Can get the entire coterm (and potentially part of your senior year) paid for with an RAship/CAship
Where you could go

Other cool classes

- CS106L
- CS108
- CS147
- CS193
- CS181 (Ethics in CS)
Announcements

• Calligraphy announcements
  – No late days may be used, no late submissions accepted
• Final is **on Saturday**, at 8:30AM, in **Cubberley Auditorium**
  – Wednesday and Thursday will be final review
"What is it like to major in Computer Science? Will you encourage us to double major in CS and another one?"

- CS + X
- ~2 CS courses per quarter (out of 4)
- WAYS requirements
- CS as a coterm
"What are popular career pathways in CS (other than app dev) and which ones were you interested in?"

- Instructor
- Product Manager
- Software Engineer
- Technical Manager
- Data Scientist
- Researcher
- Testing Engineer
- Reliability Engineer
- Tech Writer
- Usability Engineer (design)
• "What's one thing you wish you knew before you decided to major in CS?"
  – Better balance in courses (I took a lot of CS very early on)
  – Better idea of all the things I could do with CS
    • See previous question for enumeration
• "If you couldn't major in CS, what would you have majored in and why?"
  – Most likely: Management Science & Engineering (my coterm), maybe math
  – Most fun: film studies!
Your Questions

• "What's a better choice of course right after 106B: 103 or 107?"
  – 103 if you liked the discussion of big-Oh, choosing and designing data structures, or wanted to use better/faster algorithms
  – 107 if you're wondering "how does this work?" and like to think about what's going on behind the scenes
    • How does the stack work? How does the heap work? How are numbers stored? How are functions called?
  – Some people take both!
"Did you fail/do poorly in any of your CS classes? If so, how did that affect you moving forward in CS?"

- Yes! I failed the final project for a class that was one of my favorites I've taken at Stanford (thus doing poorly in the course overall)
- Learned from my mistakes, reached out to the instructor to learn more, and continued taking classes in that field and working with the instructor
• "How did you get into teaching? What makes a good teacher/teaching-assistant in your opinion?"
  – CS198 and section leading!
    • I've been teaching since freshman year, many of my closest friends are also section leaders
  – Willingness to invest time in and prioritize the course/students
  – Rethink and challenge everything constantly
  – Love of learning (and learning from better/more experienced teachers!)
  – Attention to detail
  – TA: making the lecturer's life as easy as possible (thanks Shreya!)
"What is your favorite part about being at Stanford?"

- The mentorship and relationships I have with faculty
  - Offer to help them with something or go to office hours. Ask them about them
- The chance to discover who I want to be
- The friendships I've made and the people who have been a part of my journey
- Teaching you all!
"Do you find that industrial/commercial CS is more or less the same as educational CS, or are they vastly different, i.e. is software engineering for company Y anything like completing assignments for CS X?"

- Job interviews use a lot of the concepts taught in 106B!
  - Backtracking/recursion, string processing, linked data structures (especially trees), graphs and graph algorithms, big-Oh
- Most similar to section problems in time/complexity
Your Questions

• "Mac or Windows – from a commercial, personal, and educational perspective?"
  – I use Mac for all three 😊
"If not Stanford, where would you recommend High School students look to pursue computer science?"

- MOOCs
- Online lecture videos
- Textbooks
Your Questions

• "What was the first big step you took from doing CS in college to pursuing CS as a career?"
  – First internship in CS
• Anything else?
Thank you!

• Huge thank you to Shreya for making the course run smoothly!
• Thank you to our fantastic SLs and course helpers for their hard work!
• Thank you to you all for your enthusiasm, patience, and dedication!