Computer Science 50
Fall 2010
Scribe Notes

Week 0 Wednesday: September 1, 2010
Andrew Sellergren

Contents

Computer Science 50 Week 0 Wednesday: September 1, 2010
Fall 2010 Andrew Sellergren
Scribe Notes

1 Announcements (0:00–1:00)

- This is CS50.
- 3 new handouts.
- Whether you're here because you saw David tear a phonebook in half at Harvard Thinks Big, you've been using HarvardCourses, or you want to hear David's voice crack at some point this semester, welcome!

2 Introduction to Computer Science (1:00–23:00)

2.1 The Famous Phonebook Example

- Chances are you hold some misconceptions about what computer science actually is. Computer science is not programming. Rather, programming is a tool of computer science that enables us to solve problems; it is a means to an end.
- The phonebook David holds in his hands is representative of a problem we might try to solve in computer science. If we want to look up the phone number of Mike Smith, we might begin by leafing through the phonebook one page at a time. However, the phonebook has 1000 pages or more, so it will take us an inordinate amount of time with this approach.
- Another more intelligent approach would be to flip to the middle of the phonebook to the letter L. Noting that S (for Smith) comes after L, we can disregard all the pages that come before the page we flipped to. To add some much-needed drama, we can literally tear the phonebook in half and throw the first half away.
- By tearing the phonebook in half, we've reduced 1000 pages to 500 pages. We've cut the problem in half. If we do this again and again, we can go from 500 pages to 250 to 125 and so on. Eventually we'll get down to a single page–the one we were searching for.
- The power of this algorithm lies in the logarithmic approach (more on that later). By dividing the number of pages we have to search in half with each step, we reduce the problem's complexity very quickly. Imagine that the phonebook were not 1000 pages, but rather 4 billion pages (on the order of a web search engine), how many steps would it take to find that one page we're looking for? Not as many as you might think: only 32.

2.2 Counting People

- Suppose we wish to determine the number of students in the lecture hall.

- The naive way would be to have one person stand in front of the lecture hall and point to each student once, saying "1, 2, 3,..." until each student has been counted.
- This will take as many steps as there are students, say, 350.
- Here's a better algorithm:
  1. Stand up.
  2. Think to yourself: "I am #1."
  3. Pair off with someone standing, add your numbers together, and adopt the sum as your new number.
  4. One of you should sit down, the other should go back to step 3.
- With this approach, we very quickly get an answer of 433. (According to the teaching fellows, the real total is 550, but, hey, who's counting?)[1] Here again we see that the "divide and conquer" approach is much more efficient than the linear approach.

2.3 HarvardCourses

- If you haven't already, check out HarvardCourses, which allows you to peruse Harvard's course catalog and find out what courses your friends are shopping. Even a complex application like this is ultimately composed of very simple programming statements. So too in programming do we take a "divide and conquer" approach, namely by dividing a large problem into much smaller parts, and conquering them one at a time.
- Some fun facts:
  - 11,782 courses on shopping lists
  - 1,384 Facebook users
  - 607 Google Calendars
  - 10 average number of courses on shopping lists
  - 82 is the maximum number of courses one student has on his shopping list
- We really do mean it when we say that CS50 will change how you approach tasks in your daily life. Just this summer, a former CS50 student was charged with parsing a very large text file and determining which patients visited which doctors and how many times. Using the same techniques he had used to solve Problem Set 7, he was able to write a program in PHP that would rip out the relevant information and insert it into a database (think of it as a fancy Microsoft Excel).

[1] The teaching fellows, quite literally.

2.4 Scratch and More

- In our first problem set, we'll work with a language called Scratch, developed by MIT's Media Lab to teach programming to children. In Scratch, you build programs by linking puzzle pieces together to form blocks of logic. These puzzle pieces are color coded to represent different types of logic. For example, orange represents anything that involves controlling sprites (the characters in your program) and purple represents anything that involves sprites sensing each other and their environment.
- To see what is possible with Scratch, check out Raining Men, implemented by a former student!
- If you focus on a single sprite at a time, you can begin to see how Raining Men was implemented. The singer sprites must be controlled by some sort of loop that moves them back and forth across the screen and possibly detects where the edges of the screen are. Detecting the edges of the screen might take the form of an if condition, namely, "if touching the edge of the screen, turn 180 degrees and keep moving."
- Your program doesn't have to be a static animation either. Check out Scratch Scratch Revolution, also created by a former student.
- Beyond Scratch, a number of other CS50 projects have outlived the course. Check out i saw you harvard, created by a former student as her final project.

3 The Course (23:00–50:00)

3.1 Expectations

- attend all lectures and sections
- submit nine problem sets
- take two quizzes
- design a final project

3.2 Grades

- pass/fail (David did!)
- letter grade

Please do not subscribe to the misguided notion that taking the course pass/fail somehow makes you less adequate. We at CS50 wholeheartedly encourage you to take the course pass/fail if you think it will be a more enjoyable experience that way.

3.3 Statistics

- 72% of the people in this theater have no prior programming experience
- at the outset, 43% of CS50 students describe themselves as being "less comfortable" with the material and computer science in general, 13% describe themselves as being "more comfortable," and 44% describe themselves as being "somewhere in between"
- 34% of you (or hopefully more!) out there are women

3.4 Problem Sets (Fall 2009)

- *Problem Set 0: Scratch.* Implement your very own program in Scratch.
- *Problem Set 1: C.* Implement your first program in C.
- *Problem Set 2: Crypto.* Encrypt a message using a specific cipher.
- *Problem Set 3: Game of Fifteen.* The interactive puzzle.
- *Problem Set 4: Sudoku.* A step up in UI design.
- *Problem Set 5: Forensics.* Recover a series of JPEGs from a formatted flash drive. Also solve a murder mystery.
- *Problem Set 6: Mispellings.* Write the most efficient spellchecker you can!
- *Problem Set 7: C$50 Finance.* Implement a website that allows users to manage an online stock portfolio.
- *Problem Set 8: Mashup.* Combine Google News and Google Maps.
- *Final Project.* The best part of the course! Showcase your masterpiece at the CS50 Fair.

3.5 Sections, Office Hours, Walkthroughs

- work through problems in smaller groups in sections
- get individual help in OHs
- learn how to begin each problem set at walkthroughs
- CS50's staff this year is 67 strong! We're here to help you! Come by The Lounge (our personalized teaching space with whiteboards as well as gaming consoles and Ceiling Cat)!

3.6 Course Videos and Syllabus

- available along with word-for-word transcriptions at cs50.tv
- check out a week-by-week breakdown of the course material in the syllabus[2]

3.7 Hackathon

- new this year!
- TBA, a coding session from 8 p.m. to 6 a.m. with pizza, Chinese food, and an IHOP trip
- modeled after similar events at MIT, Facebook, and Google[3]

3.8 Quotes

- demanding, but definitely doable
- social, but educational
- a focused topic, but broadly applicable skills
- CS50 is the quintessential Harvard course
- "I planned on taking the class as a freshman, then convinced myself otherwise because I "couldn't fit it into my workload," same story for soph, and junior year. I finally took it senior year cause I wouldn't get another chance, even though it still doesnt fit in with the workload haha."
- "I saw you [at the CS50 Fair]. I went to your station to see your program. You are my best friend, but when our hands touched as you handed me a smiley face sticker, my heart skipped a beat."
- "I saw you... at the CS50 fair, you were with your friends, wearing a tight white polo, looking hawt."
- "I saw you... walking around the CS50 fair with a nametag that said you were in the class. You are smart and adorable, and all this time I've assumed you were dumb and adorable. You just because so much hotter."

3.9 Staff

- check out our small army of TFs, CAs, and more
- a special welcome back to Cansu Aydede '11 and Yuhki Yamashita '11, our fearless head TFs

[2]Note that we follow the computer science convention of counting up from 0, not 1. Thus, this is Week 0.
[3]Ask me about working here!