Computer Science 50
Fall 2011
Scribe Notes

Week 0 Wednesday: August 31, 2011
Andrew Sellergren

Contents

Computer Science 50 Week 0 Wednesday: August 31, 2011
Fall 2011 Andrew Sellergren
Scribe Notes

1 Announcements and Demos (0:00–1:00)

- This is CS50.
- 3 new handouts.
- 77% of the people around you have no prior experience in computer science! Don't be afraid!
- There are a lot of stereotypes of students in computer science: that they're all nerds, that they're all men. These stereotypes are becoming misconceptions. Last year, women made up 37% of the CS50 class!
- 46% of CS50 students consider themselves among those "less comfortable."
- Welcome to our 87 staff members! A special welcome to Matthew Chartier and Rob Bowden, our head teaching fellows. If you ever have a problem during the semester and you're not sure who to reach out to, contact Matt or Rob at heads@cs50.net.
- The first ever CS50 Puzzle Day, sponsored by Facebook, will take place this Saturday! If you're interested, sign up at cs50.net/register. Form a team with at least two members and up to four and compete to win gift certificates. A raffle for a Wii will also be held.
- Dinner with CS50 and Facebook will happen this Friday night! RSVP at cs50.net/rsvp.

2 Introduction to Computer Science (1:00–45:00)

2.1 The Famous Phonebook Example

- The phonebooks David and the other CS50 staff members hold in their hands are 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 M. Noting that S (for Smith) comes after M, 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, 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, after 10 tearings or so, 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: several hundred.
- Here's a better algorithm:
  1. Stand up and think to yourself: "I am #1."[1]
  2. Pair off with someone standing, add your numbers together, and adopt the sum as your new number.
  3. One of you should sit down, the other should go back to step 2.
- With this approach, we very quickly get an answer of 640. (According to the teaching fellows, the real total is 570, but, hey, who's counting?)[2] Here again we see that the "divide and conquer" approach is much more efficient than the linear approach. Whereas counting by ones or twos would have taken us some 600 and 300 steps, respectively, the algorithm above took us only 9 or 10 steps.

2.3 More Real World Examples

- The real Wimbledon pits 128 of the best tennis players in the world against each other and produces a single winner in 7 rounds. How many rounds would it take if all 7 billion people in the world participated? Only about 33 (2^{33} is roughly 8.5 billion)!
- Computer science depends on a programmer's ability to tell a computer exactly what to do. Problems with software and hardware arise when the instructions given to a computer are not perfectly accurate or precise. Think of a computer like a bad chef following a recipe to make an omelette. If the first step read "Put egg in bowl," the computer would simply throw a raw egg, shell and all, into the bowl, which is certainly not what the recipe intended.

[1] Technically, *I'm* #1, but I'll let it slide this time.

[2] The teaching fellows, akshully.

- In contrast to these imprecise omelette recipe instructions, the instructions for filing taxes are actually quite precise (although confusing). The PDF forms that are eventually submitted to the IRS give very specific, detailed instructions about what lines to write values on and when to add them up. These instructions are comparable to those that a programmer would give to a computer.

- Think it's easy? Try it for yourself. Take a few minutes and write some instructions for making a peanut butter and jelly sandwich. Now we'll actually execute those instructions as a computer would. Here's an example of such an algorithm:

  1. Take two slices of bread from the loaf.
  2. Apply two tablespoons of jelly on both halves.
  3. Apply two tablespoons of peanut butter in the same way.
  4. Join the two halves.
  5. Have fun!

  What's the result of following this algorithm as a computer would? A big mess, unfortunately. Consider that we never told the computer to open the plastic packaging of the loaf or to use the knife to apply the jelly and peanut butter. While this might seem like a ridiculous visual, it's suggestive of the way in which we have to think in order to write computer programs. More than once this semester, you will struggle with the difficult task of translating tasks as you see them in your mind into instructions that a computer can understand and follow to achieve them.

- If you had the choice between $1 million today or receiving 1 cent today, 2 cents tomorrow, 4 cents the day after tomorrow, etc. for a month, which should you choose? Assuming you value money, you should take the second option. In a matter of 31 days, you'll bank over $10 million![3]

- The flipside of this line of thinking is when you don't choose an efficient algorithm and your program takes an inordinate amount of time to execute. We've all experienced the frustration of a spinning beach ball or hourglass, either of which might be an indication that some aspect of a program was poorly designed.

- The concept of six degrees of separation is an extension of this line of thinking. It might seem outrageous, but a connection between you and, say, Kevin Bacon[4] can be found within your friends of friends of friends of friends of friends of friends. When you do the math, however, this doesn't seem so outrageous. If you have 100 Facebook friends and each

[3] Actually, we're not even calculating the sum here. On the 31st day alone, you would receive over $10 million.

[4] I know him personally, so now you're only 1 degree of separation from him.

of your friends has 100 Facebook friends and each of their friends has 100 Facebook friends, and so on, then within 6 degrees of separation, the number of people that are connected to each other is 1 trillion (100^6). That's way more than the number of people on earth.

2.4 HarvardCourses, Shuttleboy, and APIs

- 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.
- Some fun facts:
  - 25,223 courses on shopping lists
  - 2,567 Facebook users
  - 7 is the average number of courses on shopping lists
  - 125 is the maximum number of courses one student has on his shopping list
  - 3 p.m. to 7 p.m. and 7 a.m. to 10 a.m. are the two most popular times to be "busy"
- Another app you may have used before is Shuttleboy, a shuttle schedule lookup tool. On the web or on the phone, Shuttleboy allows you to find the departure times of the next shuttles that are heading to your destination.
- HarvardCourses and Shuttleboy are examples of the types of applications you will be empowered to make by course's end. In fact, you might even build something like them for your final project. Luckily, to do so, you'll have the benefit of many different APIs, or application programming interfaces. APIs are simply frameworks that allow you as a programmer to make use of someone else's code. If you want your application to make outgoing calls and react to user input on a phone, for example, you can use the HarvardVoice API that Shuttleboy uses. No need to reinvent the wheel!

3 The Course (45:00–69:00)

- David himself once sat where you were sitting and found himself somewhat overwhelmed with the idea of taking a class in computer science. I daresay that you might one day be lecturing on computer science on the stage in Sanders![5] Actually, though, that's not the intent of the course at all. We're not out to make computer science concentrators out of all of you. Rather, we'd be extremely happy if you were taking this course just so you could find a better way to analyze that large data set for your thesis

[5] Seriously, it's not that hard. I happen to know that David spends most of his time playing Wii Tennis.

in biology or to send out an e-mail to 1000 members of your student group. We simply want you to be able to use these wonderful tools called computers to their full advantage.

3.1 Expectations

- attend all lectures and sections
- submit nine problem sets
- take two quizzes
- submit 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 Problem Sets

- Problem Sets are the main work of the course. They give you a chance to show off the knowledge you've gained in lecture and section and office hours in order to create programs and applications that actually *do* something. Ultimately these should be challenging, but also fun. As Rob mentioned, it's pretty cool that week on week, the finished product of your work for this course will be a game that your roommate can play or a website that you can show to your friends and family back home.
- Problem Sets come in two editions: standard and Hacker. If you want a little extra challenge out of these projects, try the Hacker Edition.
- You have a certain number of late days that will be provided to you for turning in problem sets after the deadline with no penalty. We understand that life happens and you might not have time for that problem set when a midterm rolls around in another of your courses. Make use of your late days and buy yourself some time.
- The lowest score of your problem sets will be dropped, as well, but see the syllabus for fine print.

3.4 The Unofficial Guide to CS at Harvard

- Oh, the places you'll go! Check out the Unofficial Guide to CS at Harvard for different paths you can take after CS50 and Harvard.

3.5 CS50 Appliance

- This year, for the first time, we'll be transitioning away from using remote servers to using your own computers for compiling and executing your programs. With the CS50 Appliance, you'll have the benefit of a virtual machine which allows you to run an operating system boxed within your computer's native operating system. Just double click the icon and suddenly you've launched an instance of Linux!

3.6 Office Hours

- Office Hours are being rebooted this year! In recent years, the sheer volume of students that have shown up to Office Hours have made them overwhelming. We're hoping to fix this this year by moving Office Hours to the dining halls to coincide with Brain Break. Instead of a handful of TFs, we will have a dozen or so on duty at all times from 9 p.m. to midnight on Mondays through Thursdays.

- When you attend Office Hours this year, you'll need only login to cs50.net and virtually raise your hand. Your question will be displayed to the TFs and it will be answered in the order it was received relative to other questions. We even have a fancy iPad app that will allow the CS50 Greeter to match staff members and students, alerting students when it's their turn with a TF.

3.7 Lectures

- This year for the first time, students at Harvard Business School, Belmont High School, and Watertown High School will be taking the class virtually. Welcome, guys!

- Check out a week-by-week breakdown of the course material and problem sets in the syllabus.[6]

3.8 Walkthroughs, Course Videos, The Lounge

- Learn how to begin each problem set at Walkthroughs on Sunday nights. When you find yourself completely lost after reading the specification PDF and you are staring at a blinking cursor on your screen (hopefully not on Thursday night), come to the Walkthrough (or watch the video) which will jumpstart you.

[6] Note that we follow the computer science convention of counting up from 0, not 1. Thus, this is Week 0.

- Course videos will be available online along with word-for-word transcripts in case you miss a live lecture or you just don't want to get out of bed before 1 p.m.!
- Come hang out with us at the CS50 Lounge, stocked with a Wii, a foosball table, candy machines, and even Ceiling Cat!