

122 Review & Starting Algorithm Analysis

Fall 2017 - Aaron S. Crandall, PhD

Today's Outline

- Announcements
- Thing of the Day
- Linux final discussion
- Reviewing last stages of 122
- Intro to algorithm analysis

Announcements



- We have our TAs. I'm making the final open labs schedule
 - Starts next week. Should have standing times M-Th
 - We have 2 grading TAs and 2 open lab TAs I'll get a schedule onto blackboard and such.
- Cougs in Space next meeting: Wednesday 5:30pm in Spark 223
 - <https://www.facebook.com/cougsinspace/>
- EECS Student Club Night: Tuesday 4:10pm in Sloan 175

Thing of the Day: Origin of the webcam

- The webcam was invented at Cambridge University to monitor a coffee pot in the Trojan Room.
- 128x128 greyscale camera on a video capture card
- Connected to the World Wide Web in about 1993
- Lasted until 2001
 - The last coffee pot was sold back to Krups, where it was refurbished
 - Now kept at the German Museum of Technology in Berlin
- https://en.wikipedia.org/wiki/Trojan_Room_coffee_pot



Last class recap

- We discussed:
 - Linux
 - Linux conventions & general OS ideas
 - Started using g++ & Make to build software on Linux
- Any questions on this, or microassignment #1 (Hello world)?
- I should have the next microassignment out later this week
 - It's time to start coding!

A quick poll about recursion

- Recursion!
 - How much have people used it before?
 - Get your base case correct FIRST
 - Will be used for many data structures in this course
- Recursion has a very natural flow for many data structures
 - Notably, trees, sorting, and graphs
- What's the big feature of how recursion operates?
(it's got a data structure that it relies upon, if it's not explicit)

Reviewing CptS 122 - At least the last bits

- Structures:
 - Vectors vs. Arrays (kinda lists)
 - Linked List
 - Stack
 - Queue
 - Binary search tree
- How many items do these have in them? How would we represent that?
 - Also, how much space do they take up?

Aspects of Chapter 1

- Mathematics Concepts to Have: logarithms, series, modulus, proofs (induction, counterexample, contradiction)
- Recursion: definition, uses, memory structure, how it truly works in C++, use for induction proofs, design rule (all calls work):
 - Base Case: One case in the function can be solved without recursion
 - Making Progress: Each call gets sequentially closer to base case
 - Design Rule: Assume all recursive calls work
 - Compound Interest Rule: Never duplicate work by doing same instance twice

C++11 Features (at least a few of them)

- Separation of interface and implementation
- vector and string
- Lvalues, Rvalues, and References
- `std::swap` & `std::move`
- Big Five: why and when do you need to do them?
 - Destructor
 - Copy constructor
 - Move constructor
 - Copy Assignment operator=
 - Move Assignment operator=

Starting algorithms & analysis

- How can we compare two algorithms that achieve the same result?
 - What's common to both algorithms?
- We actually compare the function growth rates for algorithms
 - It turns out algorithms are functions, so we can formalize them
 - Then comparison is based on how they change with regards to input size
- Wednesday:
 - Proofs as a basis for analyzing the accuracy of algorithms
 - C++/C++11 review and features
 - Diving into formal definitions of Big-O