CS4102 Algorithms
Fall 2019

Alan Turing
Ada Lovelace
Edsger Dijkstra
Al-Khwarizmi
Robbie Hott
Robert Tarjan
David Wu
Gauss
Tony Hoare
Donald Knuth
Stephen Cook
What is an algorithm?

• In mathematics and computer science, an algorithm is a self-contained sequence of actions to be performed. Algorithms can perform calculation, data processing and automated reasoning tasks. [Wikipedia Aug 2018]

• In mathematics and computer science, an algorithm is an unambiguous specification of how to solve a class of problems. Algorithms can perform calculation, data processing and automated reasoning tasks. [Wikipedia Jan 2019]

• In mathematics and computer science, an algorithm is a set of instructions, typically to solve a class of problems or perform a computation. Algorithms are unambiguous specifications for performing calculation, data processing, automated reasoning, and other tasks. [Wikipedia Aug 2019]

• An algorithm is a step by step procedure to solve logical and mathematical problems. [Simple English Wikipedia Aug 2019]

• Motivating example
How much concrete do I need?

Need an accurate approximation

\[ \pi \]

4.3km (2.7mi) diameter
\pi = 3.14159265359...

Circumference = 2\pi
\[ \pi = 3.14159265359... \]

Perimeter > \(2\pi\) > Perimeter
\[ \pi = 3.14159265359... \]

\[ 2\pi > \text{Perimeter} = 6 \]
π Approximation Algorithm

Solve for $x$

$$x = \frac{2}{\sqrt{3}}$$

$$\pi = 3.14159265359... \quad 1 \text{ digit correct}$$

$$\frac{12}{\sqrt{3}} = \text{Perimeter} > 2\pi > \text{Perimeter} = 6$$

$$3.46 > \pi > 3$$
\[ \pi = 3.14159265359... \quad \text{3 digits correct} \]

\[ 6 + \frac{20}{70} = \text{Perimeter} > 2\pi > \text{Perimeter} = 6 + \frac{20}{71} \]

\[ 3.14285 > \pi > 3.14084 \]
How to analyze this approach?

• How fast do we “converge?”
• How much work is needed to do better?
Another Algorithm

- Look up and explain the solution for extra credit!
Better $\pi$ Approximation (Ramanujan)

\[
\frac{1}{\pi} = \frac{2\sqrt{2}}{9801} \sum_{k=0}^{\infty} \frac{(4k)! (1103 + 26390k)}{(k!)^4 396^{4k}}
\]

\[
\pi = 3.141592653589793238462643383279502884197169399375105820974944592307816406286208998628
\]

$k = 0$
\[
\pi \approx 3.1415927
\]

8 digits per iteration!

$k = 1$
\[
\pi \approx 3.1415926535897938
\]
Goals

- Create an awesome learning experience
- Instill enthusiasm for problem solving
- Give broad perspective on Computer Science
- Have fun!
Warning

• This will be a very difficult class
  – Hard material
  – “Holy Grail” of computer science
  – Useful in practice
  – Job Interviews

• Lots of opportunities to succeed!
  Hopefully not you...

I Quit!
• **Very difficult course.** Besides the grading/homework thing though, I really loved the course and I loved Hott.

• Algorithms can be a boring subject, but Hott introduces complex problems in class which makes it fun to follow along with. I thoroughly enjoyed the homework assignments although they were VERY difficult. It was nice to collaborate with my friends. We actually had some fun doing the assignments.

• I just have so say, I both love and hate this class at the same time. I hate that this class seemed to take up my life and I saw my group members and TA's more than my friends this semester, but I LOVE the professors who taught this class. They both have a similar sense of humor and love for puzzles, have their own quirky humor to liven up class, and just make class somewhere you can't hate to be.

• You can tell that he really cares about students and their understanding of the material and doesn't want to torture students too much.

• Although I hated discrete math and theory of computation, this class was one of my favorites I have taken as a CS major.

• It is a great course.
While difficult, students have done well...
Office Hours

- Rice 210
  - Poll time!  [www.menti.com](http://www.menti.com)  code: 34 29 54
  - By appointment
Requirements

- Discrete Math (CS 2102)
- Data Structures (CS 2150)
- Derivatives, series (Calc I)
- Tenacity
- Inquisitiveness
- Creativity
- No textbook required
- Highly recommended:
  - Polya. *How to Solve It.*
Grade Breakdown

- Homework: 60%
- Midterm: 20%
- Final: 20%
- Extra Credit: 10%
Homework

• 11 assignments total
• Mix of written and programming assignments
• Written:
  – 2/3 of all assignments
  – Must be typeset in LaTeX (tutorial is HW0)
  – Submit as a pdf and a zip folder containing tex file and any supplements
    • Submissions without both attachments (pdf, zip) will not be graded
• Programming:
  – 1/3 of all assignments
  – Must implement in Python or Java
Homework 0

- Homework 0 is out!
  - Learning LaTeX
  - You MUST submit both:
    - A zip with your tex and image
    - A PDF of the final document
  - Due next Tuesday (but don’t wait that long!)
Academic Integrity

• Collaboration Encouraged!
  – Groups of up to 5 per assignment
  – List your collaborators (by UVA computing ID)

• Write-ups/code written independently
  – DO NOT share written notes / pictures
  – DO NOT share documents (ex: Overleaf)

• Be able to explain any solution you submit!

• DO NOT seek published solutions online
Late Policy

\[ \text{grade} = \text{grade}_{\text{earned}} e^{-\frac{1}{2\phi} \text{days}} \]

- Exponential decay, accepted until solutions posted
- Extra credit: find a radioactive isotope with half-life closest to your homework's
Exams

• Midterm
  – October 15
  – In-class / take-home hybrid

• Final
  – Registrar’s official date/time (COMBINED)
  – Monday, December 9, 7-10pm
Regrades

- Conducted in person with course staff
  - Time and Location: TBD
  - By appointment
Extra credit

• Given for extraordinary acts of engagement
  – Good questions/comments
  – Quality discussions
  – Analysis of current events
  – References to arts and music
  – Extra credit projects
  – Slide corrections
  – Etc. Just ask!

• Email: extra.credit.cs4102@gmail.com
Feedback

• I am not a course dictator, I am a civil servant
• I’m open to any suggestion to help you learn
• Let me know!
  – In person
  – Email
  – Piazza
Attendance

• How many people are here today?
• Naïve algorithm
  – Everyone stand
  – Professor walks around counting people
  – When counted, sit down
• Run time?
  – Class of $n$ students
  – $O(n)$
• Other suggestions?
Better Attendance

1. Everyone Stand

2. Initialize your “count” to 1

3. Greet a neighbor who is standing: share your name, full date of birth (pause if odd one out)

4. If you are older: give “count” to younger and sit. Else if you are younger: add your “count” with older’s

5. If you are standing and have a standing neighbor, go to 3