CS 4102: Algorithms
Lecture 1: Introduction and Logistics

David Wu
Fall 2019
What is an Algorithm?

According to Wikipedia... algorithms are
• a self-contained sequence of actions to be performed
• an unambiguous specification of how to solve a class of problems

Motivating example

Takeaway: Being unambiguous is not always easy!
A Historic Perspective

Euclid (300 BC)

Al-Khwarizmi (800)

Gauss (1600)

Ada Lovelace (1800)

Stephen Cook (1900)

Alan Turing

Edsger Dijkstra

Don Knuth
A Concrete Example

Need an accurate approximation \( \pi \)

How much concrete do I need?

4.3km (2.7mi) diameter
$\pi$ Approximation Algorithm

$\pi = 3.14159265359...$

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

\[ 2\pi > \text{Perimeter} = 6 \]
Solve for $x$

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

$$\pi = 3.14159265359...$$

1 digit correct

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

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

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

\[ 3.14285 > \pi > 3.14084 \]
Some Questions to Ask?

How fast do we “converge?”

How much work is needed to do better?

Using this approach, we get $\frac{1}{2}$ digit of $\pi$ with each additional side – to get $n$ digits, we need a polygon with $2n$ sides.
Another Algorithm

https://youtu.be/HEfHFsfGXjs

Extra credit: Look up and explain the solution!
Better $\pi$ Approximation
Better $\pi$ Approximation
(Ramanujan Series)

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

$\pi = 3.14159265358979323846264338327950288419716939937510582097494...$

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

8 digits per iteration!

$k = 1$
$\pi \approx 3.1415926535897938$
Better \( \pi \) Approximation
(Ramanujan Series)

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

\[\pi = \boxed{3.14159265358979323846264338327950288419716939937510582097494\ldots}\]

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

8 digits per iteration!

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

Ramanujan series are the basis for fastest algorithms for computing \( \pi \)
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!
• He is very reasonable in what he expects from students and makes the course appropriately hard and time consuming.
• The professor was extremely helpful both in and out of class and does not appear to give hard homework for the sake of defeating his students.
• The vast majority of the time he was able to explain the hard concepts effectively and made me more motivated and confident about the rigorous topics during the semester.
• The class is very difficult. It's not easy, but it's important material.
• The homework felt brutally difficult compared to the lecture and midterm, but was probably so for a reason.
• This class was the perfect balance of being true to the difficult course material but entirely fair
• The class was pretty difficult, but I felt like I was given more than enough resources to succeed (and do better than I actually did lol)
• He was also incredibly flexible with his assignments knowing full well that his assignments were incredibly difficult
• The homework was very difficult and I think a little too hard to get an A in
• GETTING AN A IN THIS CLASS IS HARDER THAN SNEAKING INTO THE BASE OF NSA WITH OR WITHOUT GETTING SHOT
While Difficult, Students Have Done Well...
Office Hours

Prof. Hott  
Rice 210

Prof. Wu  
Rice 501


See Google Calendar for TA office hours
Logistics

All course materials available on Collab

Course website: 
https://www.cs.virginia.edu/~jh2jf/courses/fall2019/cs4102/

Piazza: https://piazza.com/class/jz4i8z5w7ym5mr
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.*

No textbook required

Highly recommended:

Polya. *How to Solve It.*


Freely accessible online via the UVA library
Grade Breakdown:

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

Write-ups/code written independently
- DO NOT share written notes
- 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:** name the radioactive isotope with half-life closest to your homework
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
  • Piazza
  • Email (dwu4@virginia.edu)
How many people are here today?

Naïve algorithm
  • Everyone stand
  • Professor walks around counting people
  • When counted, sit down

Complexity?
  • Class of $n$ students
  • $O(n)$ “rounds”

Other suggestions?
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 count

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

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

Requires $O(\log n)$ “rounds”