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]
What is an algorithm?

• In mathematics and computer science, an algorithm is a finite sequence of well-defined, computer-implementable instructions, typically to solve a class of problems or to perform a computation. Algorithms are unambiguous specifications for performing calculation, data processing, automated reasoning, and other tasks. [Wikipedia Jan 2020]

• 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

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

\[ \pi = 3.14159265359... \]

Circumference = \( 2\pi \)
\( \pi \) Approximation Algorithm

\[
\pi = 3.14159265359...
\]

Perimeter > 2\( \pi \) > Perimeter
π Approximation Algorithm

π = 3.14159265359...

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

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

Solve for \( x \)

\[ x = \frac{2}{\sqrt{3}} \]

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

\[ 3.46 > \pi > 3 \]
$\pi$ Approximation Algorithm

$\pi = 3.14159265359...$ 3 digits correct

$6 + \frac{20}{70} = \text{Perimeter} > 2\pi \approx 6 + \frac{20}{71}$

$3.14285 > \pi > 3.14084$
How to analyze this approach?

- How fast do we “converge?”
- How much work is needed to do better?
Better $\pi$ Approximation (Ramanujan)

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

$$\pi = 3.141592653589793238462643383279502884197169399375105820974944592\ldots$$

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

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!
You suffer a lot and learn a lot! Professors are very knowledgeable and they make sure that students understand the material during lecture.

While this was an extremely difficult course, I learned a great deal and going to office hours really helped supplement that as well.

Great teacher: his lectures make murky material clear, and he's very willing to talk outside of class to help students. Great class: the material is fun, although I'm not getting an A in this class :(

Great, challenging class.

So much homework. It took so long.

Overall, a great class. A must-take!

Hard class but worth all the time spent in it

Professor Brunelle was one of the best CS professors I have had. He made the content very easy to digest and was throughout the semester it felt as though he truly cared for our grades. I appreciated how many resources he gave us in order to allow us to succeed, even though the class is difficult.

Homeworks were very hard but incredibly rewarding and helped a lot to understand the materials. Professor was inspiring and displayed a great understanding of the material.

I learned a lot but I will probably get a bad grade because the assignments were tough as nails.
While difficult, students have done well...

https://vagrades.com/uva/CS4102
Who Am I?

University of Virginia
University of Virginia
University of Virginia
Requirements

• Discrete Math (CS 2102)
• Data Structures (CS 2150*)
• Series (Calc I)
• Tenacity
• Inquisitiveness
• Creativity

* Enforced pre-req of C- or better.
Lectures

• Live Lectures
  – Tuesday/Thursday at 2pm
  – Recording posted after

• Live Workshops
  – Completely optional to attend
  – Tuesday/Thursday at 12:30pm
  – Recording posted after
Textbook

• Highly recommended:

Tasks

- Exercises
  - Approximately 2 per unit

- Quizzes
  - 1 per unit
  - You can re-take
Exercise 0

- Exercise 0 is out today
  - Intended to get you ready for future problem sets
    - LaTeX
    - Programming
  - Worth Bonus points
Academic Integrity

• Collaboration Encouraged!
  – Groups of up to 5 per assignment (you + 4)
  – List your collaborators (by UVA computing ID)
• Write-ups/code written independently
  – DO NOT share written notes / pictures / code
  – DO NOT share documents (ex: Overleaf)
• Be able to explain any solution you submit!
• DO NOT seek published solutions online
Late Policy

• By default, late submissions not accepted
• If something comes up that prevents you from submitting quality work on time, let me know what’s going on
Exams

• No exams
Regrades

- Conducted using the submission system:
  - Submit within 5 days of receiving your grade
  - Request a regrade if the rubric was misapplied
Course webpage

- www.cs.virginia.edu/~njb2b/cs4102/su20
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!

• Submit to me via email

• Will be used for qualitative grade adjustments
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 office hours
  – Email
  – Piazza
Office Hours


- TA office Hours:
  - Hosted Via Discord
  - Begin on Thursday

- Instructor Office Hours:
  - Hosted Via Zoom
  - Begin immediately
Where does it end?

• I have a pile of string
• I have one end of the string in my hand
• I need to find the other end
• How can I do this efficiently?
Rope End Finding

1. Set aside the already obtained end

2. Separate the pile of rope into 2 piles, note which connects to the known end (call it pile A, the other pile B)

3. Count the number of strands crossing the piles

4. If the count is even, pile A contains the end, else pile B does
Why does it work?
How efficient is it?

• $T(n) = \text{count}(n) + T\left(\frac{n}{2}\right)$
• $T(n) = 4.5 + T\left(\frac{n}{2}\right)$
• Base case: $T(1) = 1$
Let’s solve the recurrence!

\[ T(1) = 1 \]
\[ T(n) = 4.5 + T\left(\frac{n}{2}\right) \]
\[ 4.5 + T\left(\frac{n}{4}\right) \]
\[ 4.5 + T\left(\frac{n}{8}\right) \]
\[ \ldots \]
\[ 1 \]

\[ T(n) = \sum_{i=0}^{\log_2 n} 4.5 = 4.5 \log_2 n \]