CS 4102: Algorithms

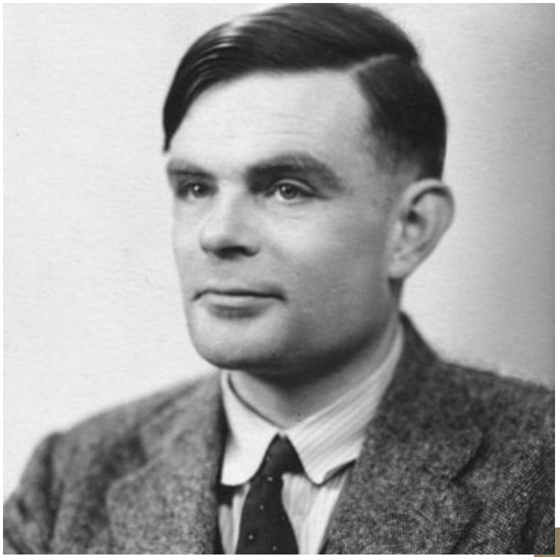
Lecture 1: Introduction and Logistics

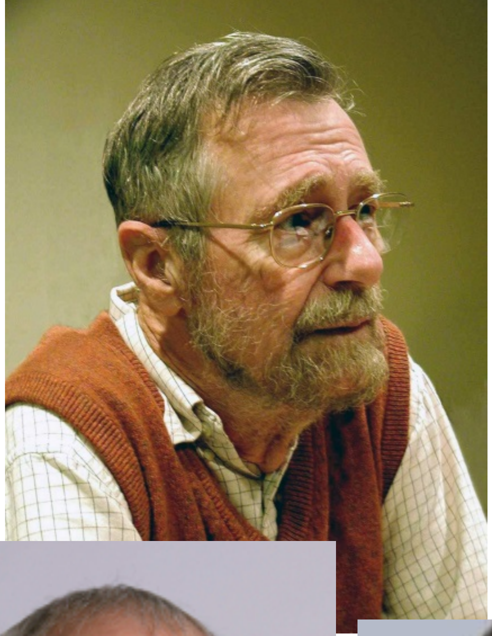
Co-instructors: Robbie Hott and Tom Horton

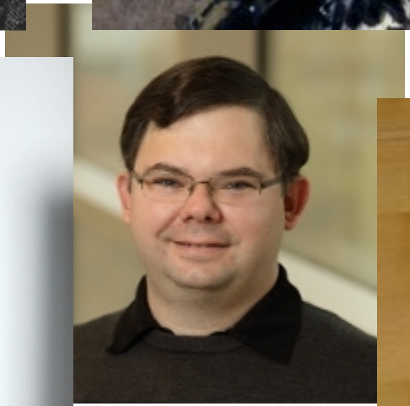
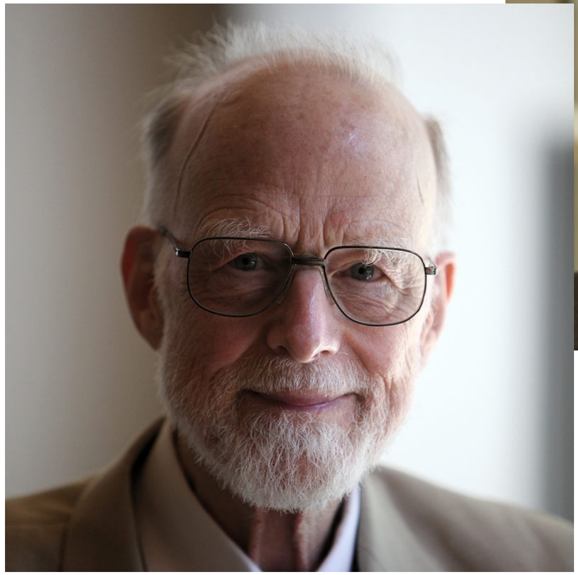
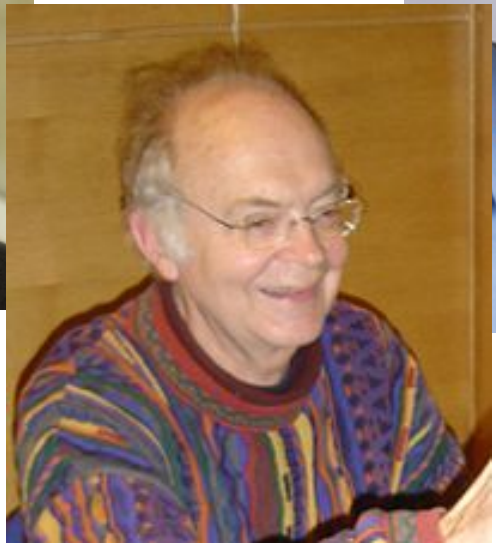
(These are slides for Horton's section)

Spring 2020

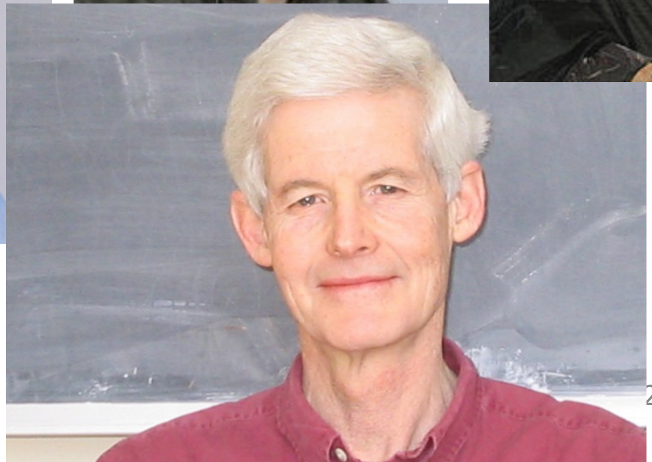
Who's Who in Algorithms?





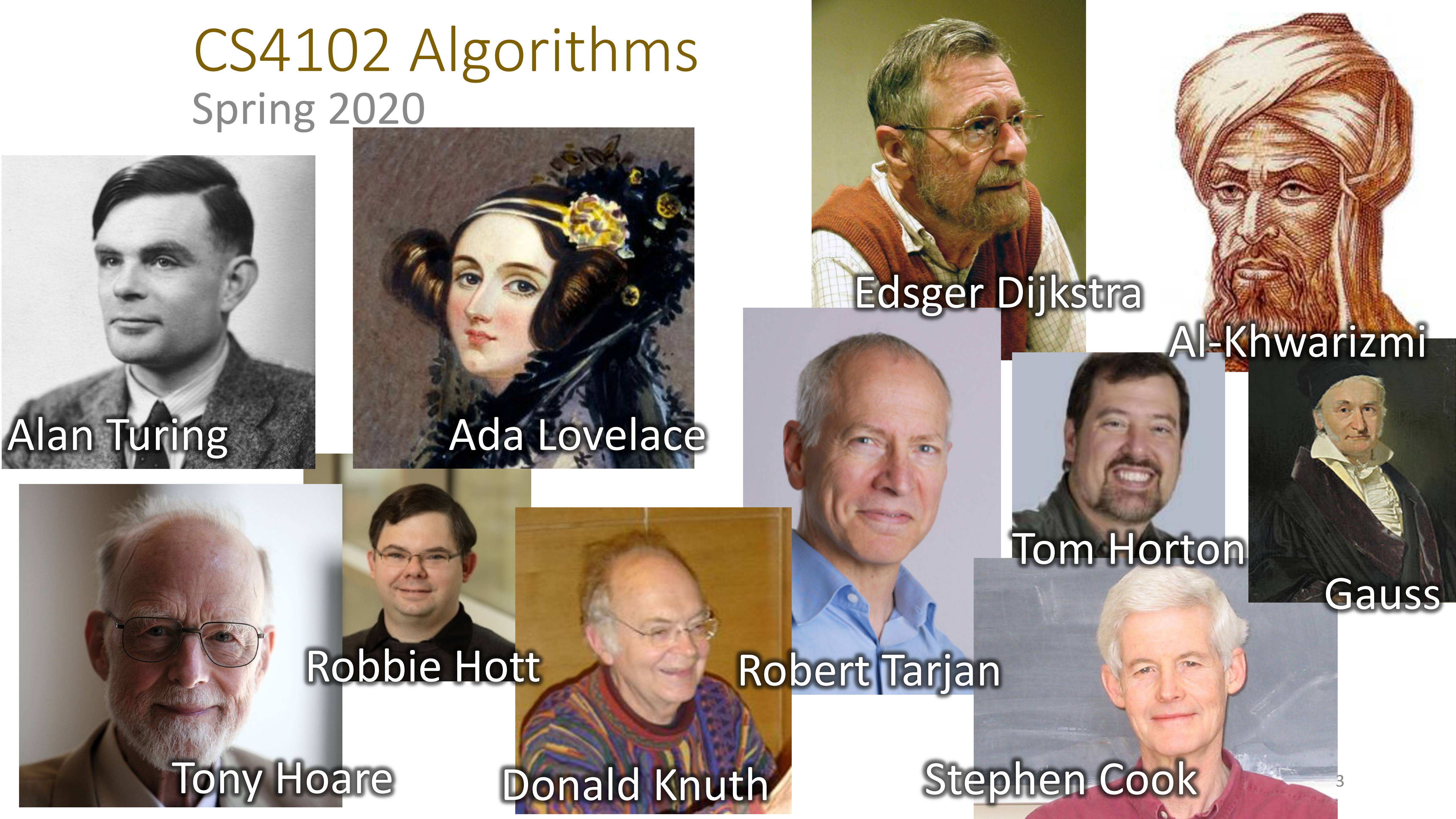








CS4102 Algorithms

Spring 2020



A Historic Perspective

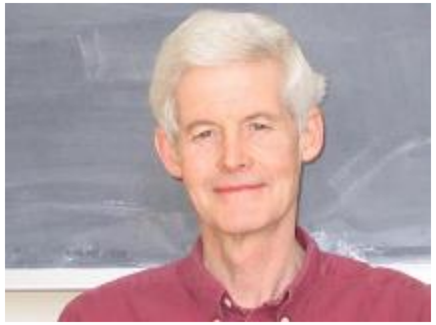
Euclid

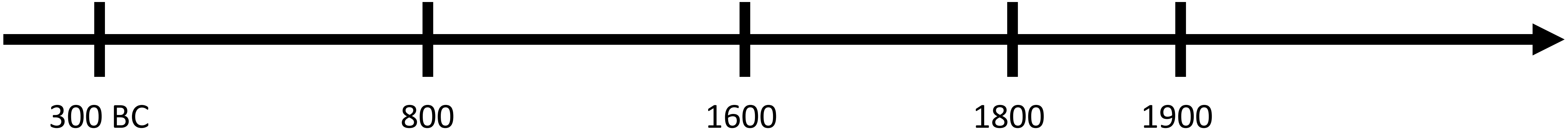
Al-Khwarizmi

Gauss

Ada Lovelace

Stephen Cook





Alan Turing

Edsger Dijkstra

Don Knuth

What Is an Algorithm?

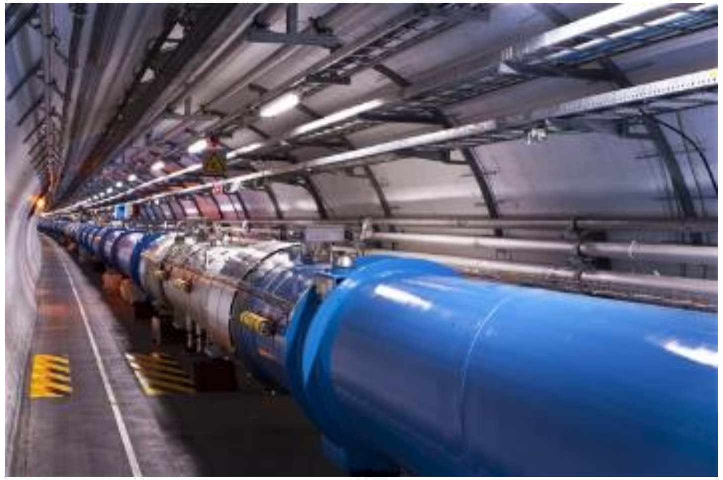
- In mathematics and computer science, an algorithm is a finite sequence of well-defined, computer-implementable instructions, typically to solve a class of problems or to perform a computation. Algorithms are unambiguous specifications for performing calculation, data processing, automated reasoning,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

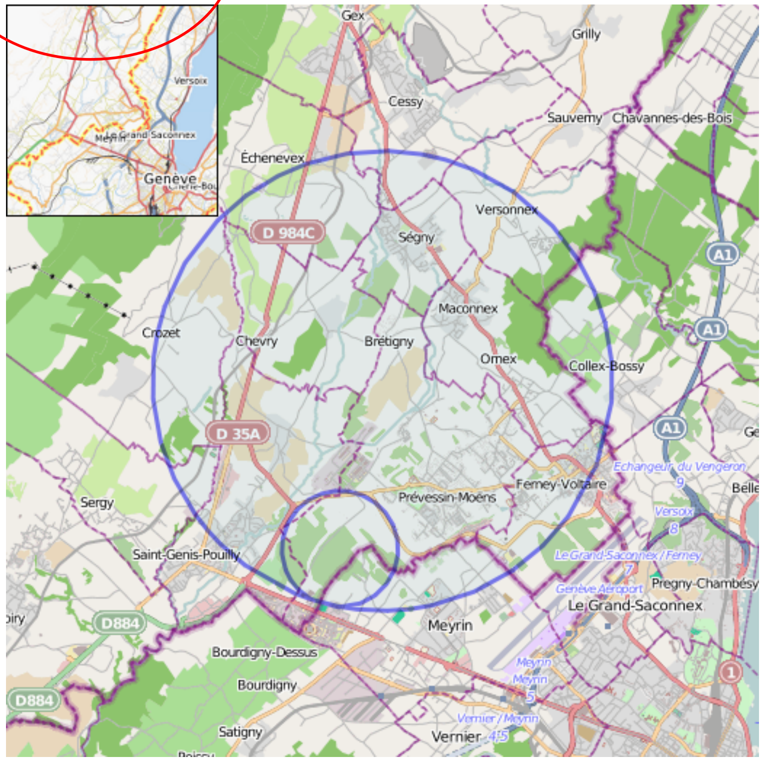
Takeaway: Being unambiguous is not always easy!

A *Concrete* Example

Need an accurate approximation → π



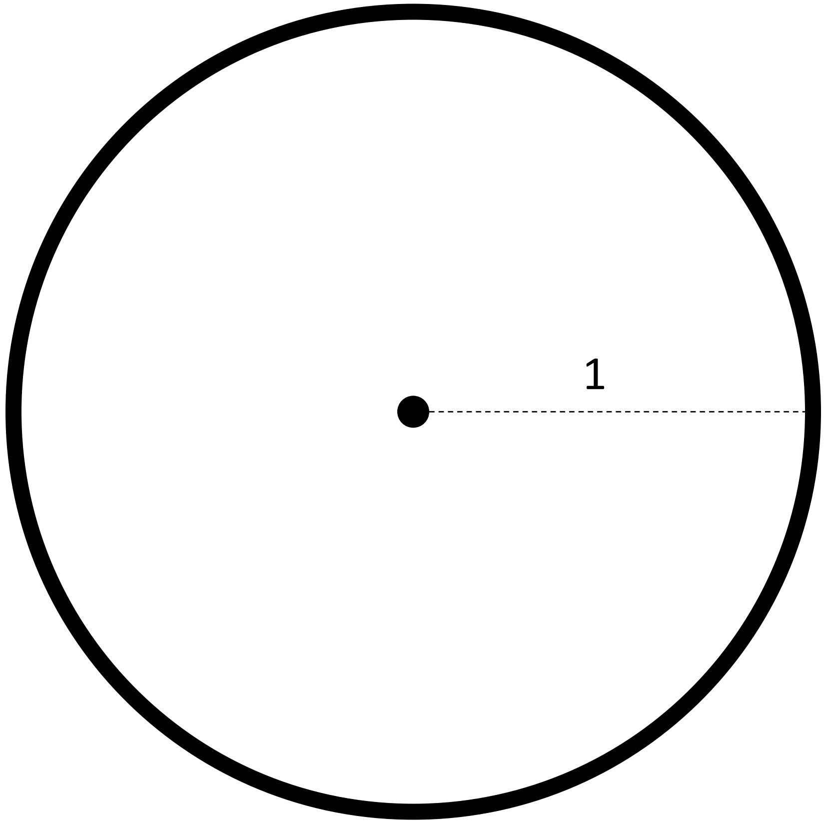
How much concrete do I need?



4.3km (2.7mi) diameter

π Approximation Algorithm

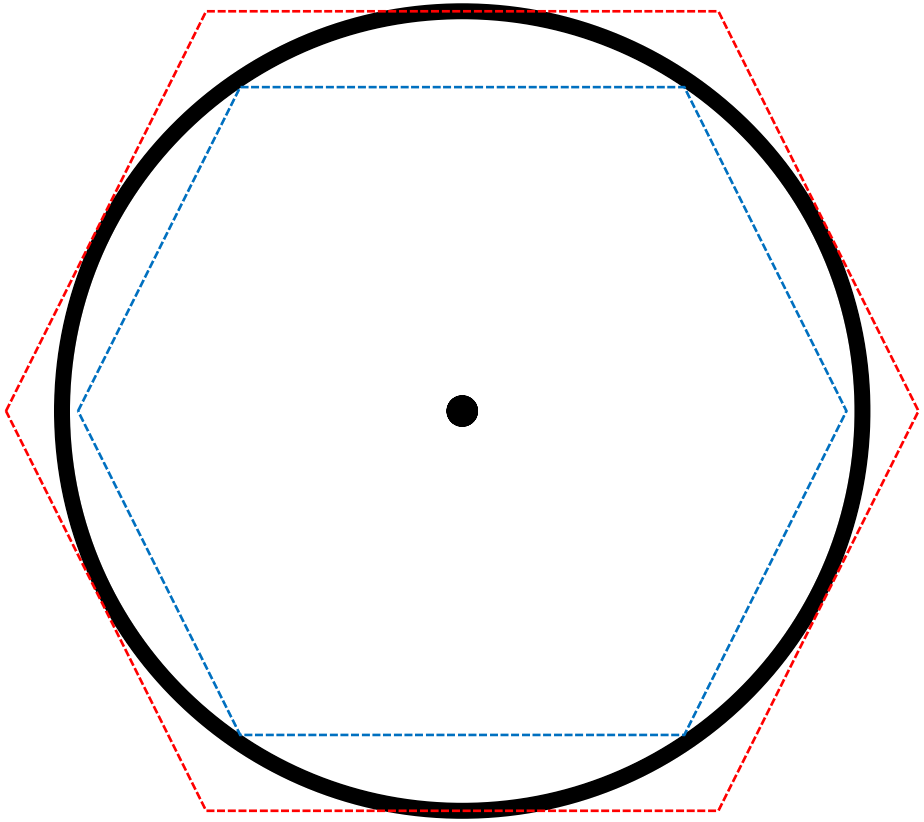
$\pi = 3.14159265359...$



Circumference = 2π

π Approximation Algorithm

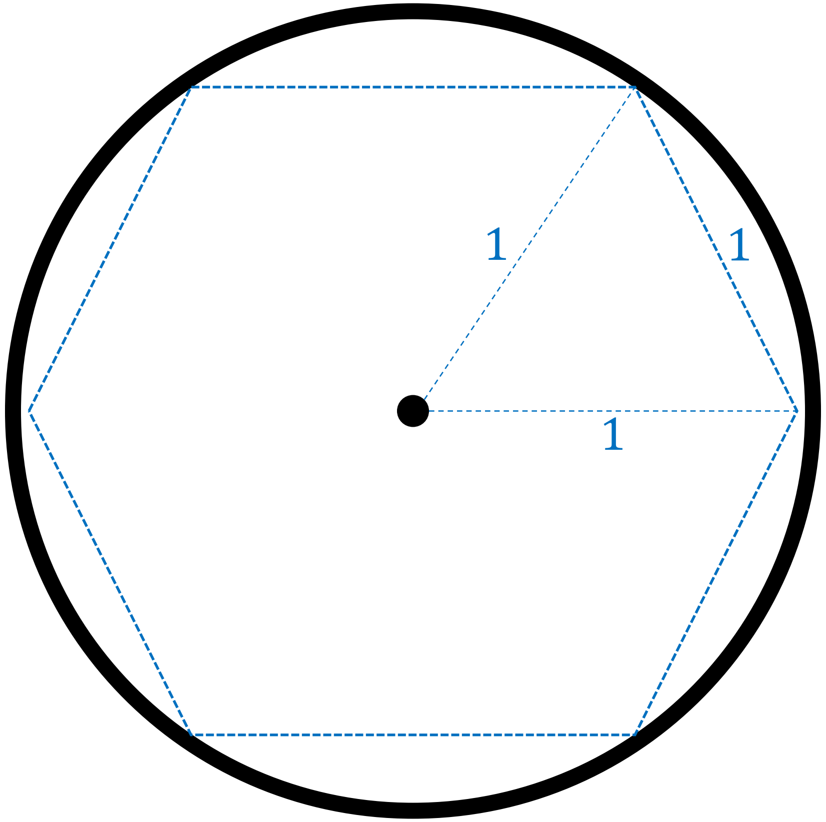
$\pi = 3.14159265359...$



Perimeter $> 2\pi >$ Perimeter

π Approximation Algorithm

$\pi = 3.14159265359...$



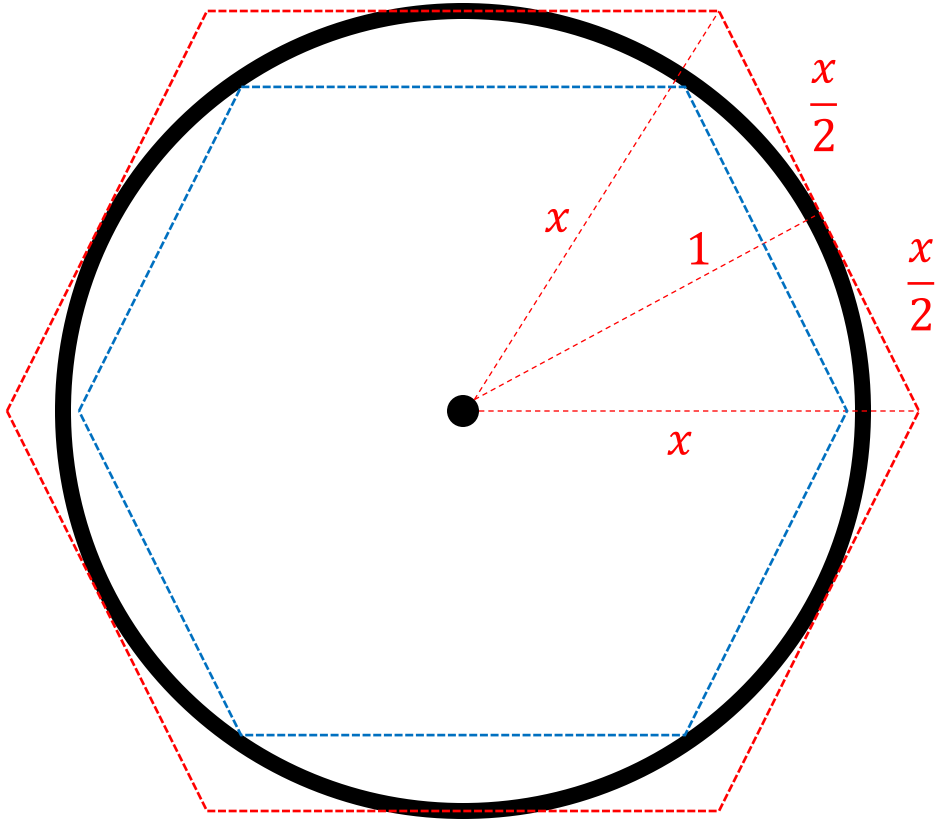
$2\pi >$ Perimeter $= 6$

π Approximation Algorithm

$\pi = 3.14159265359...$ 1 digit correct

Solve for x

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

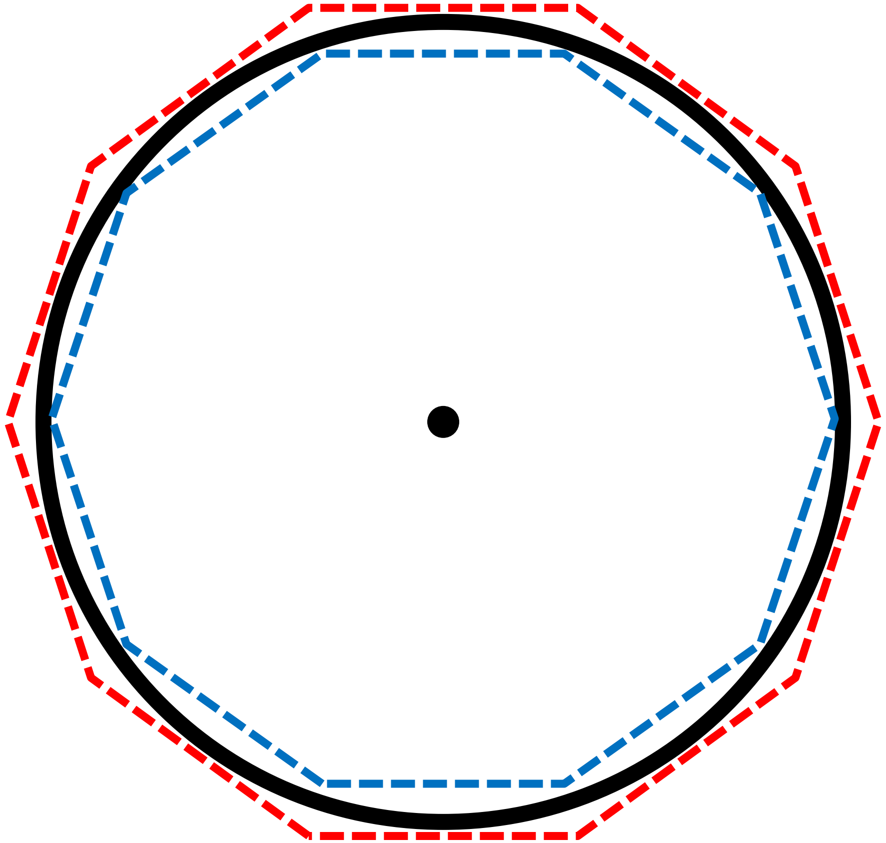


$$\frac{12}{\sqrt{3}} = \text{Perimeter} > 2\pi > \text{Perimeter} = 6$$

$$3.46 > \pi > 3$$

π Approximation Algorithm

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



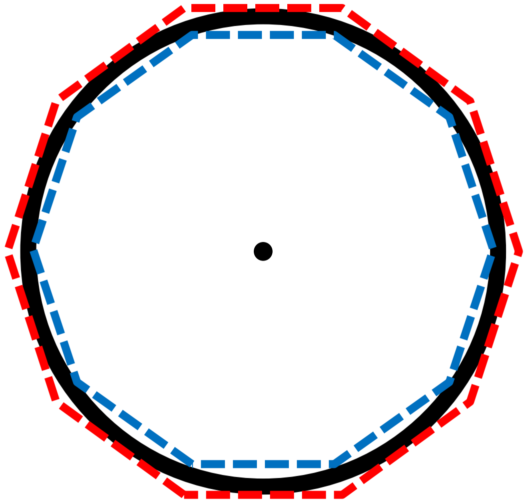
$6 + \frac{20}{70} =$ Perimeter $> 2\pi >$ Perimeter $= 6 + \frac{20}{71}$

$3.14285 > \pi > 3.14084$

Some Questions to Ask?

How fast do we "converge?"

How much work is needed to do better?



Using this approach, we get ½ digit of π 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 π Approximation

Better π Approximation (Ramanujan Series)

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

$$\pi = 3.14159265358979323846264338327950288419716939937510582097494...$$

$k = 0$

$\pi \approx 3.1415927$

$k = 1$

$\pi \approx 3.14159265358979$

8 digits per iteration!

Better π Approximation (Ramanujan Series)

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

$\pi = 3.1415926535897932384626433832795028841$

Ramanujan series are the basis for <u>fastest</u> algorithms for computing π

$k = 0$

$\pi \approx 3.1415927$

8 digits per iteration!

$k = 1$

$\pi \approx 3.141592653589793$8

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

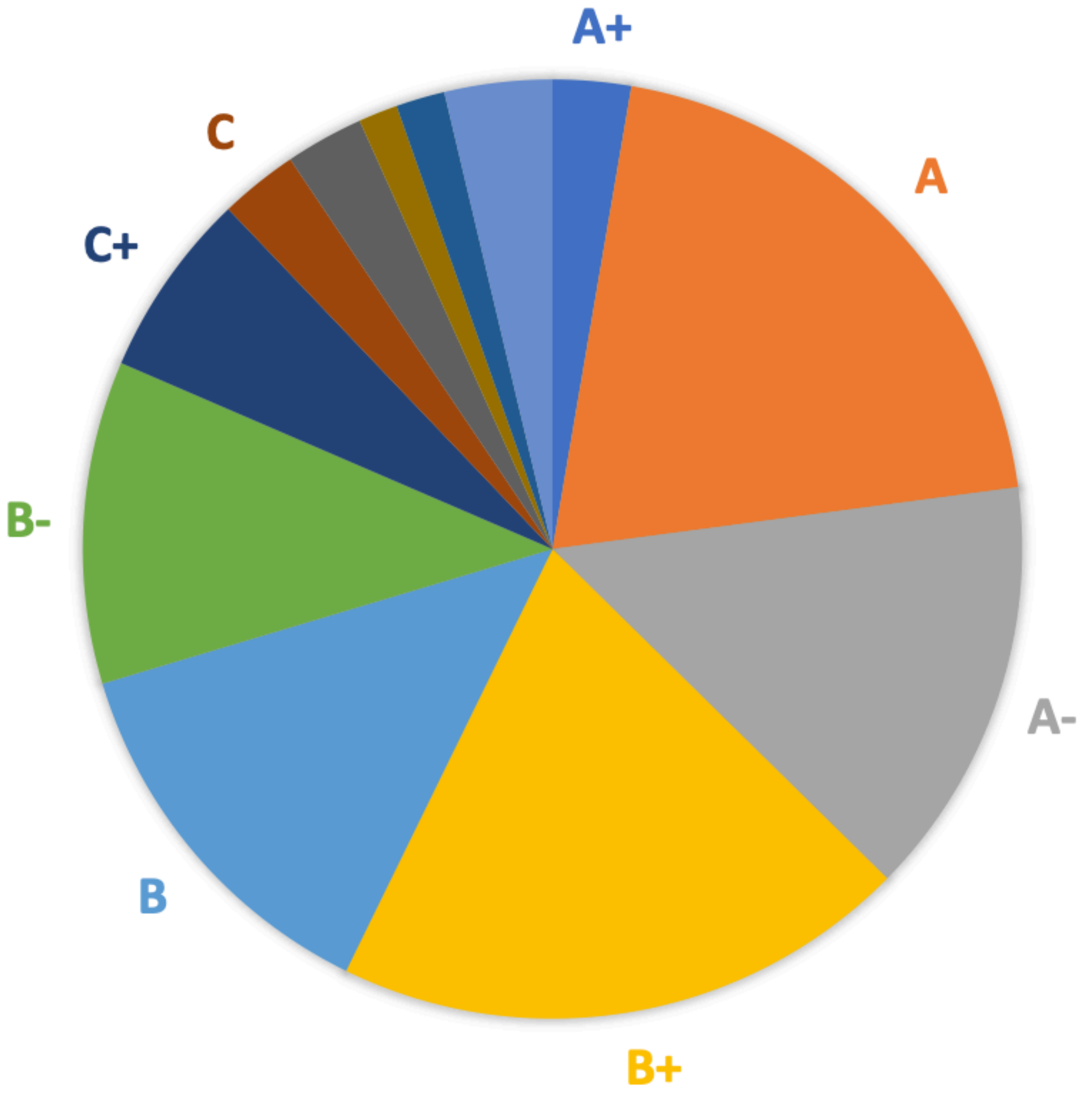
Lots of opportunities to succeed!

Hopefully not you…

Recent CS4102 Comments

- Although I hated discrete math and theory of computation, **this class was one of my favorites** I have taken as a CS major.
- Algorithms can be a boring subject, but [they] introduce 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.
- The material taught in the class is very very relevant. During my coding interviews, I was asked material that was presented in class.
- I loved this course! It honestly was not as bad as some people say with regards to content or time commitment.
- This class is just hard and time consuming.
- This was a challenging yet extremely rewarding course, as I feel that I have learned so much that I will use in my future career in computer science.

While Difficult, Students Have Done Well…



Co-Instructors

Prof. Hott

Rice 210

Prof. Horton

Rice 401

Office Hours TBD! Wait for announcement!

See Google Calendar for TA office hours

Logistics

All course materials available on **Collab**

Including syllabus, resources, slides, HWs, etc.
(See Syllabus for statements on policies, most covered here in these slides.)

Course website? Yes, see link on Collab
Piazza for student questions? Yes, through Collab
Anonymous feedback? No….

Requirements

Discrete Math (CS 2102)

Data Structures (CS 2150) with C- or higher

Derivatives, series (Calc I)

Tenacity

Inquisitiveness

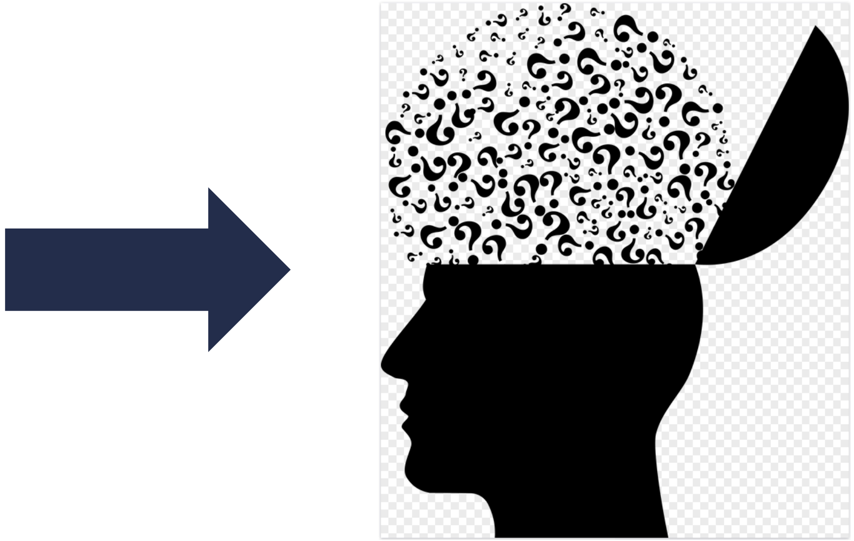
Creativity

Note: CS2150 pre-req taken seriously. Don't meet it? Need approval or you will be dropped (after add deadline). ☹

"Learning Sources"

From what sources will you learn?

What I say in Lectures
What you get from the slides
Explanations & details you read
Activities you do in/out of class
Assignments



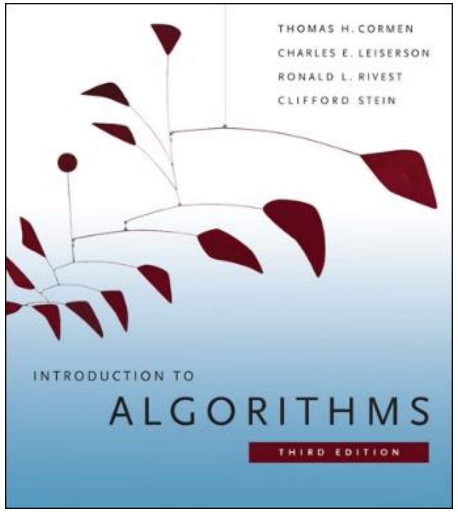
- All of these are important.
- Realistically, IMHO it's impossible to get all the "book knowledge" from lectures and slides!

Textbook

You should read and study material other than the slides.

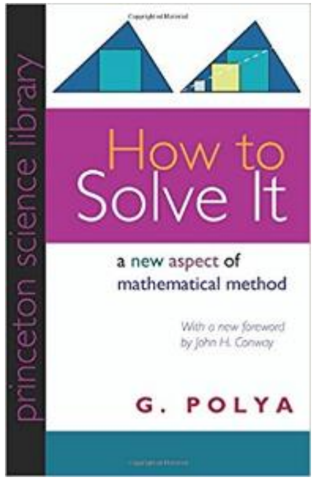
There are options, but a textbook is the easiest option.

I'll post readings from CLRS, urge you to read them or get that info from another source. **Note:** We will also have some resources posted on Collab site.



Cormen et al. (CLRS) *Introduction to Algorithms*. Third Edition.

Also recommended (but not a primary source)



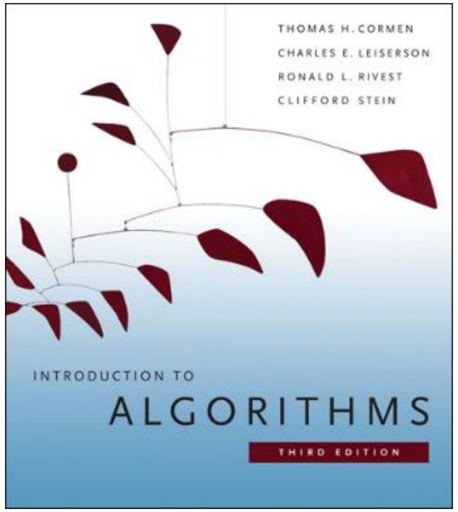
Polya. *How to Solve It.*

Textbook

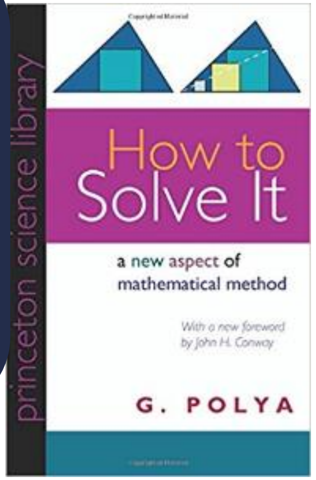
You <u>should</u> read and study material other than the slides.

There are options, but a textbook is the easiest option.

I'll post readings from CLRS, urge you to read them or get that info from another source.



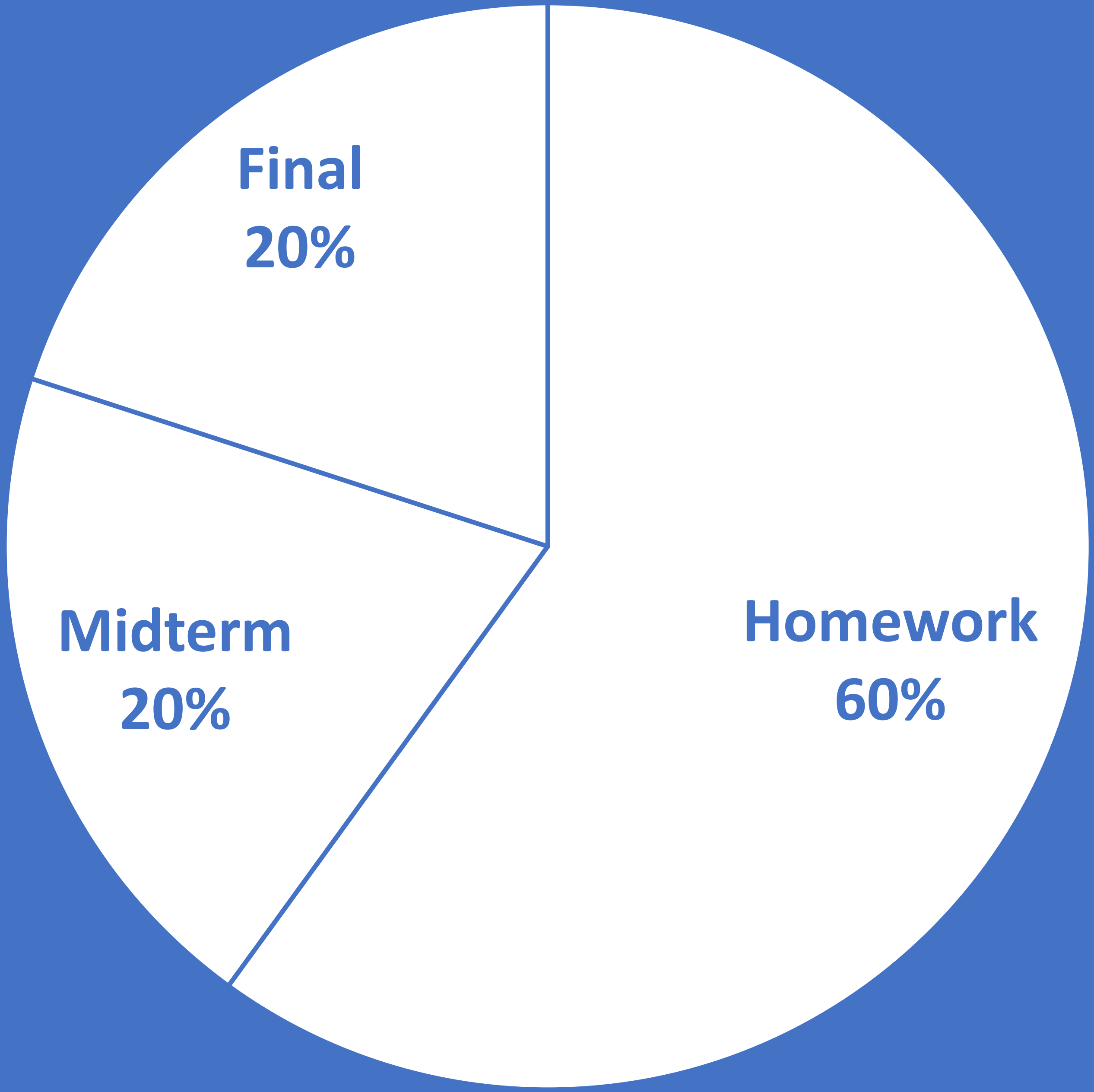
Freely accessible online via the <u>UVA library</u>



Cormen et al. (CLRS) *Introduction to Algorithms*. Third Edition.

Polya. *How to Solve It.*

Grade Breakdown



10% Extra Credit

Homework

10 assignments total

Mix of written and programming assignments

Written:

- 2/3 of all assignments
- Must be typeset in LaTeX (tutorial is HW0)
- Submit 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!

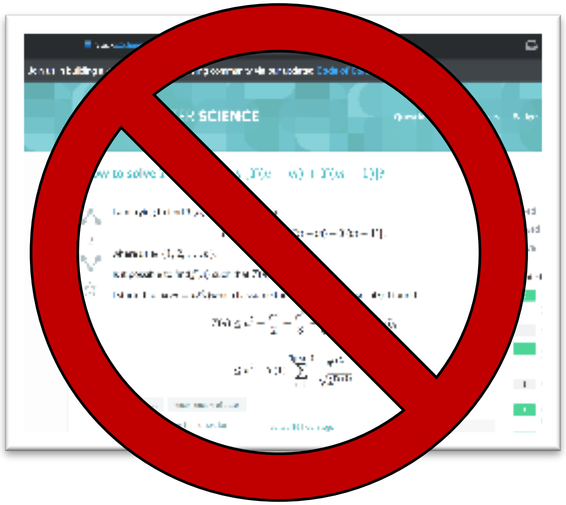
- Groups of up to 5 per assignment (you + 4)
- List your collaborators (by UVA computing ID)

Write-ups/code written independently

- DO NOT share written notes / pictures / code
- DO NOT share documents (ex: Overleaf)
- DO NOT share debugging of code

Be able to explain any solution you submit!

DO NOT seek published solutions online

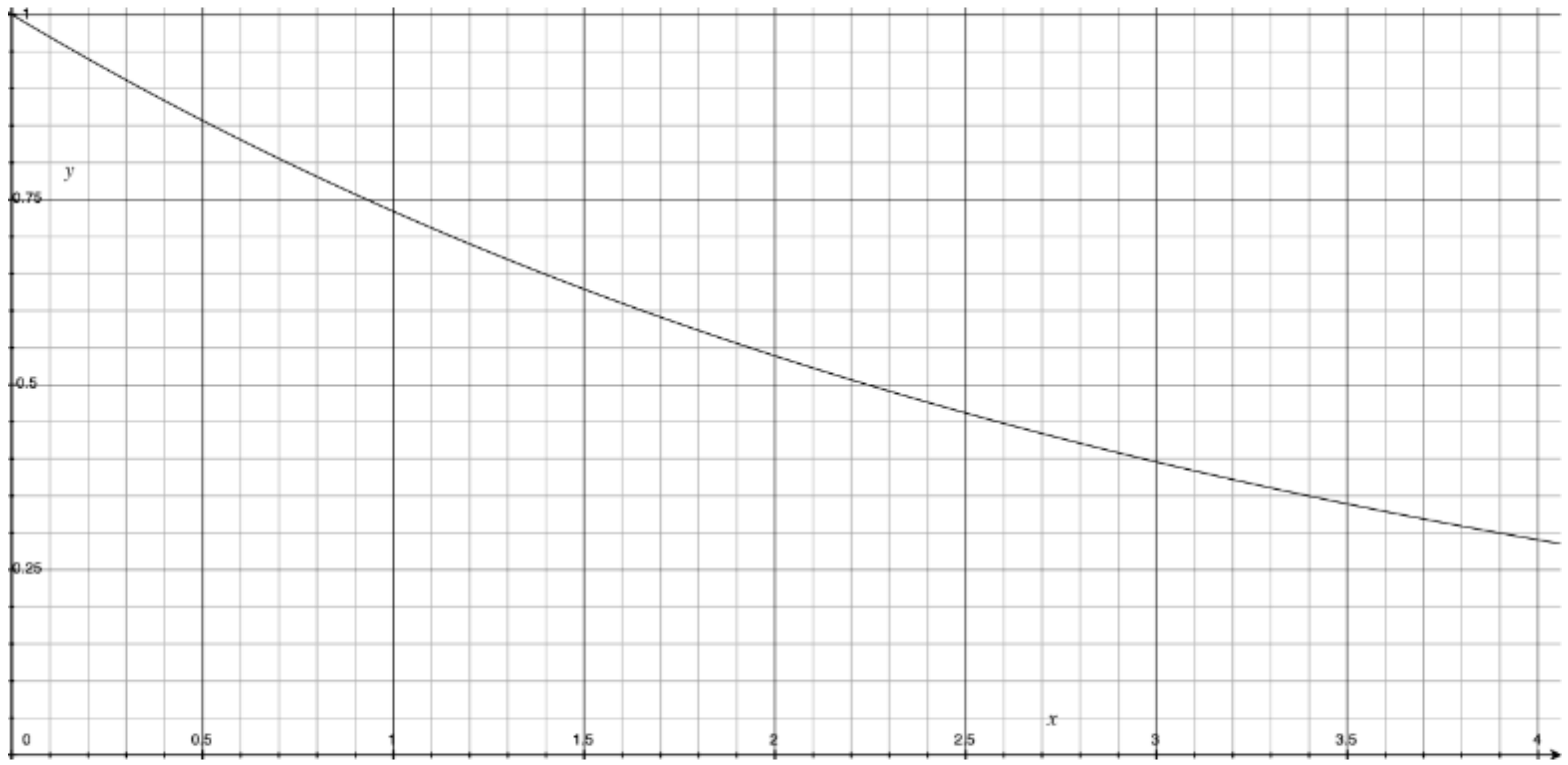


Late Policy

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

Exponential decay, accepted until solutions posted

Extra credit: name the radioactive isotope with half-life closest to your homework



Exams

Midterm

- Weds., March 4 (right before spring break)
- In-class / take-home hybrid

Final

- Registrar's official date/time (All sections COMBINED)
- Saturday, May 2, 7-10pm
- Room TBD

Regrades

Conducted in person with course staff

- Time and Location: TBD
- By appointment
- Limited to 2 weeks after grades released

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: **extracredit.cs4102@gmail.com**

Feedback

We professors are not course dictators, more like civil servants.

I'm open to any suggestion to help you learn.

Let me know!

- In person
- Piazza
- Email (horton@virginia.edu)
  - PLEASE: put CS4102 in subject line of all emails

Attendance

How many people are here today?

Naïve algorithm

- Everyone stand
- Professor walks around counting people
- When counted, sit down

Time Complexity?

- Class of n students
- $O(n)$ “rounds”

Other suggestions?

(How can we compare different approaches?)

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 count

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

Better Attendance

Requires $O(\log n)$ "rounds"

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