Portland State University

Winter'21 CS 584/684

Algorithm Design & Analysis

What is an algorithm?

In mathematics and computer science, an **algorithm** (/ˈælɡərɪðəm/ (listen)) is a finite sequence of well-defined, computer-implementable instructions, typically to solve a class of problems or to perform a computation.[1][2] Algorithms are always unambiguous and are used as specifications for performing calculations, data processing, automated reasoning, and other tasks.

Can you name a few algorithms?

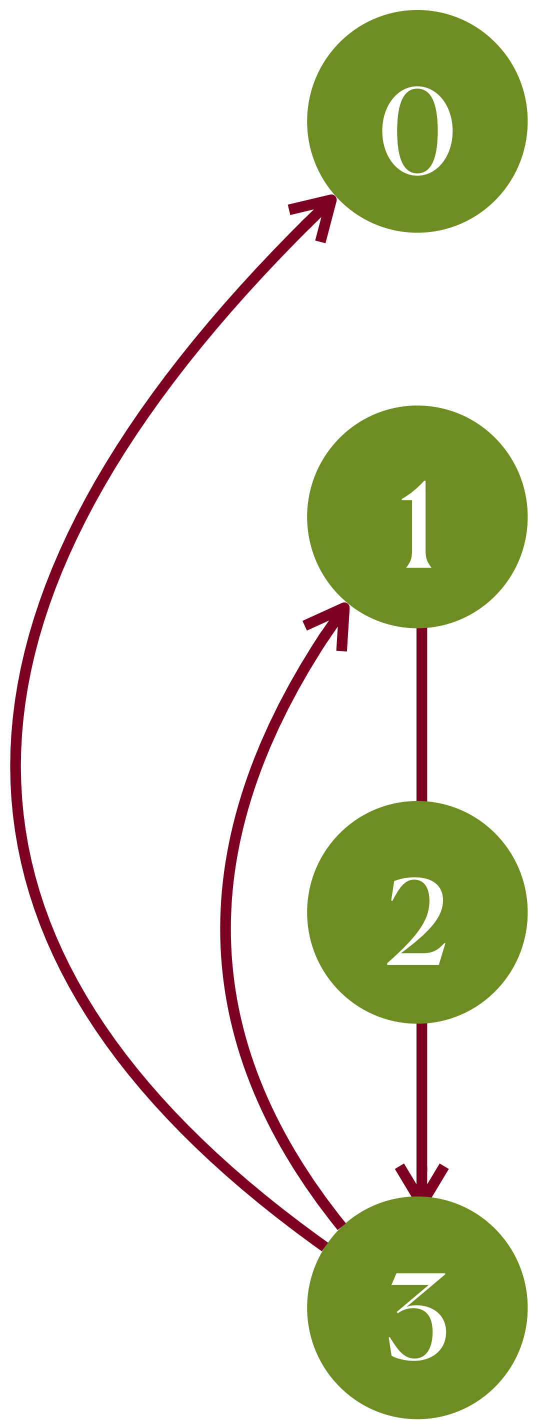
Multiplication: back to grade school

- **Long multiplication algorithm**

```
          2 0 2 1
×           3 6 5
-----------------
        1 0 1 0 5    (2021×5)
      1 2 1 2 6 0   (2021×60)
+   6 0 6 3 0 0   (2021×300)
—————————————————
      7 3 7 6 6 5
```

- Unambiguous set of instructions: above
- Solving a well-defined problem: multiplying two non-negative integers

Principal questions



What problem to solve?

Is the algorithm correct?

How much resource it costs?

Can we do better?

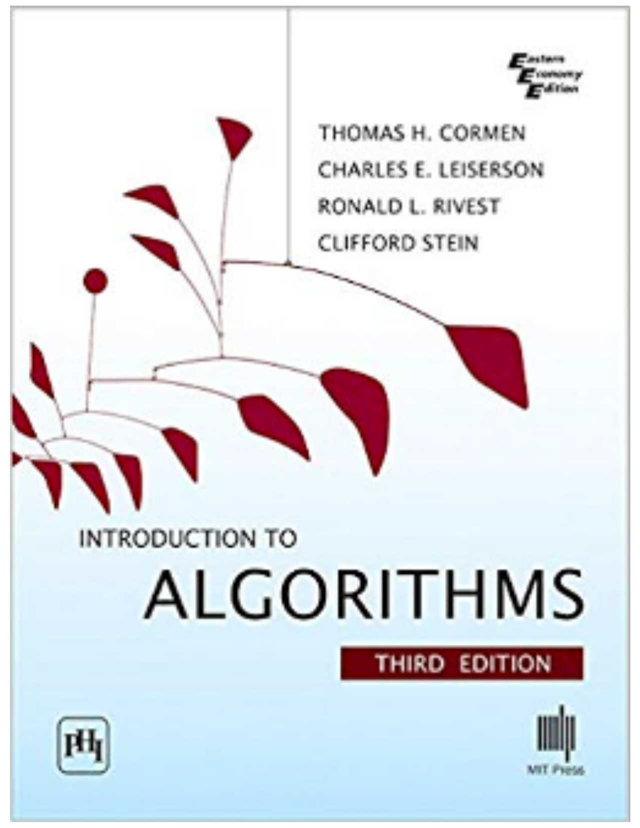
DESIGN

ANALYSIS

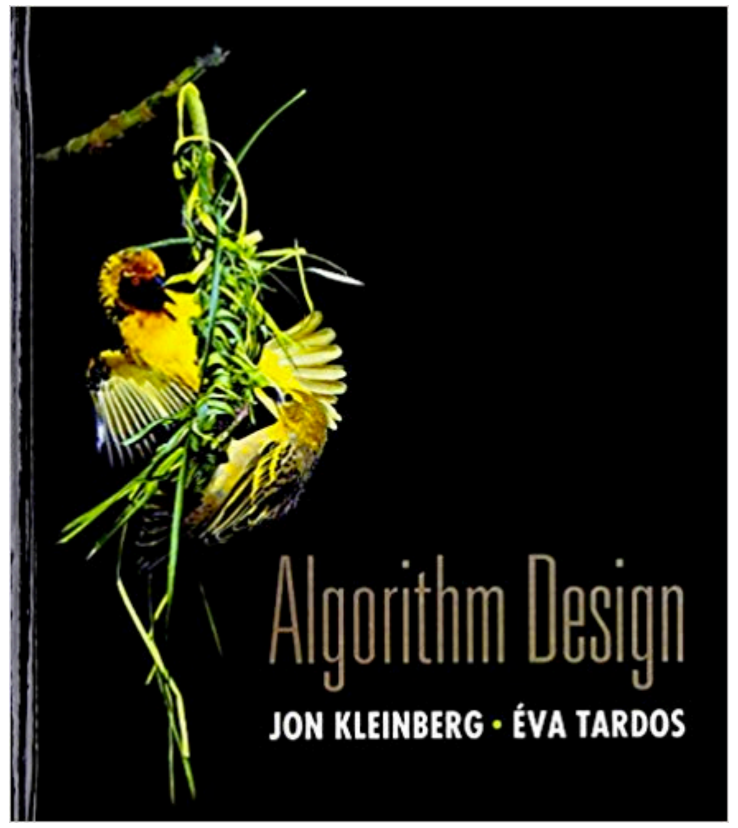
Logistics

- **Instructor**: Prof. Fang Song.
- **Email**: fsong@pdx.edu.
- **Office hours**: F 8:30 - 10am and by appointment via Zoom.
- **TA**: Steven Libby (slibby@pdx.edu).
- **Zoom links**: “PSU Classes” Calendar.

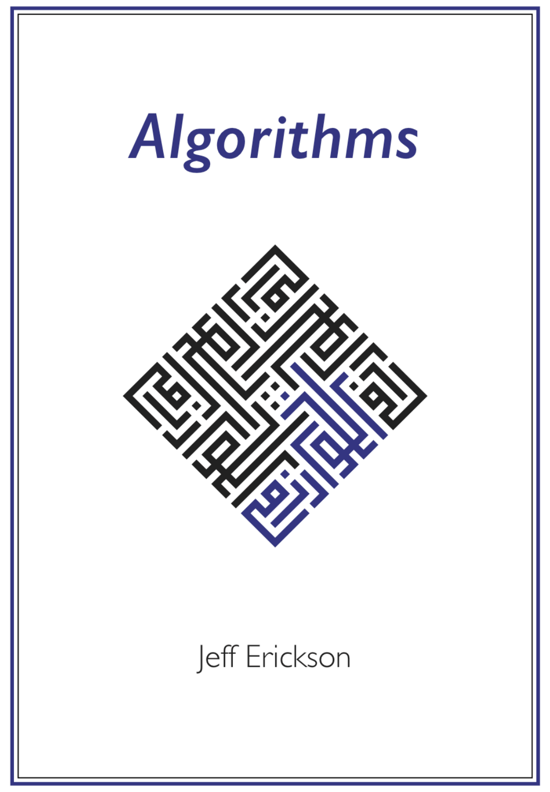
- **Recommended texts**
  - Primary CLRS: E-copy available via PSU library



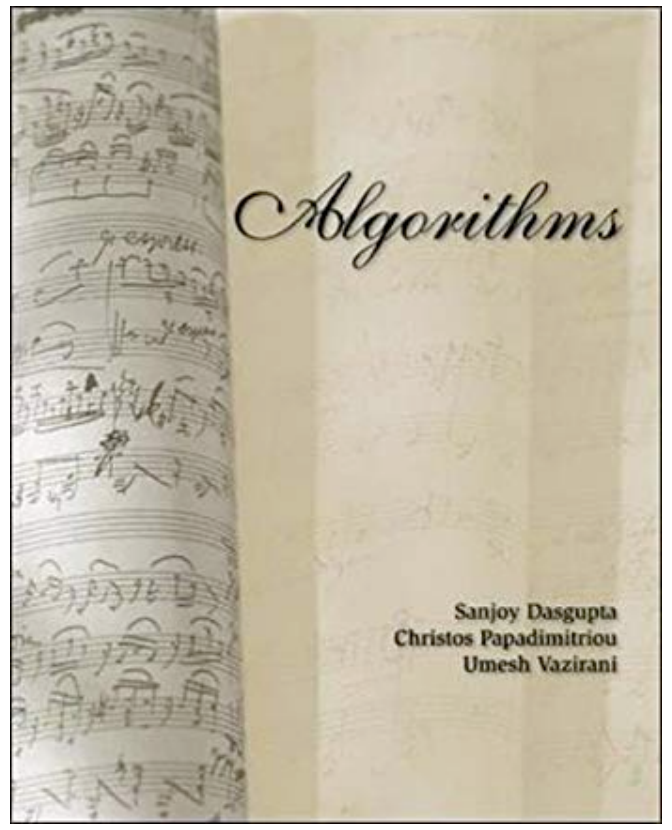
CLRS



KT



JE



DPV

Prerequisite & main topics

- **CS 350 or equivalent**
  - Basic data structures and algorithms: sorting, graph traverse
  - Math maturity: basics of combinatorics, linear algebra, probability

Comfortable with READING & WRITING mathematical Proofs

  - Uncertain? Talk with me. Not a good idea to proceed if not ready.
- **This course: more formal treatment and advanced materials**
  - Standard + selected topics at the end (quantum algorithms, etc.)

Policy

- **Final exam:** 30%.
- **Midterm exam:** 25% week 5.
- **Participation:** 5%.

- **Homework:** 40%.
  - Weekly.
  - Collaboration.
  - Find more on syllabus.

Policy cont'd

- **Academic Integrity**
  - PSU Student Code of Conduct

- **Academic accommodation**
  - Contact DRC (503-725-4150, drc@pdx.edu) and notify me.

- **Recording lectures**
  - Comply with FERPA, the Acceptable Use Policy and PSU's Student Code of Conduct.
  - Sharing outside this class not permitted.

How to succeed?

- Study the reading materials in advance.
- Start on assignment EARLY!
  - Make 20% progress per day for 5 days!
- Ask a lot of questions.
- Form study groups.

To-do #0: course webpage

You’ve probably accomplished it already. Congrats!

- Familiarize yourself with it https://fangsong.info/teaching/w21_5684_alg/
- “Schedule” page
  - Zoom links, reading materials, assignments.
- ”Resource” page contains additional materials.

Check regularly!

To-do #1: get to know each other

- It's very helpful to form a study group.

- **Gather town:** https://gather.town/app/CsidNQ8umdbRglZG/psu-w21-cs5684

To-do #2: make a choice

1. Homework management
    a. Google classroom
    b. Gradescope

2. Communication & discussion
    a. Email (via Google Groups)
    b. Slack
    c. Piazza or alternatives

Complete the survey by Friday at https://forms.gle/e56YsGkRnaiBHbYA6

Now the real meat

1. Overview of algorithm design techniques.

2. Overview of algorithm analysis.

3. Growth of functions, asymptotic notations.

4. LaTeX tutorial.

Algorithmic techniques

★ **Reduction: a meta technique, always keep in mind.**

- **Brute force**
- **Divide and conquer**
  - Decompose a problem into smaller sub-problems and compose the solutions.
- **Dynamic programming**
  - Memorize soln's to subproblems that occur repeatedly.
- **Greediness**
  - Make a local optimal choice for subproblems.
- **Randomization**
- **… Creativity**

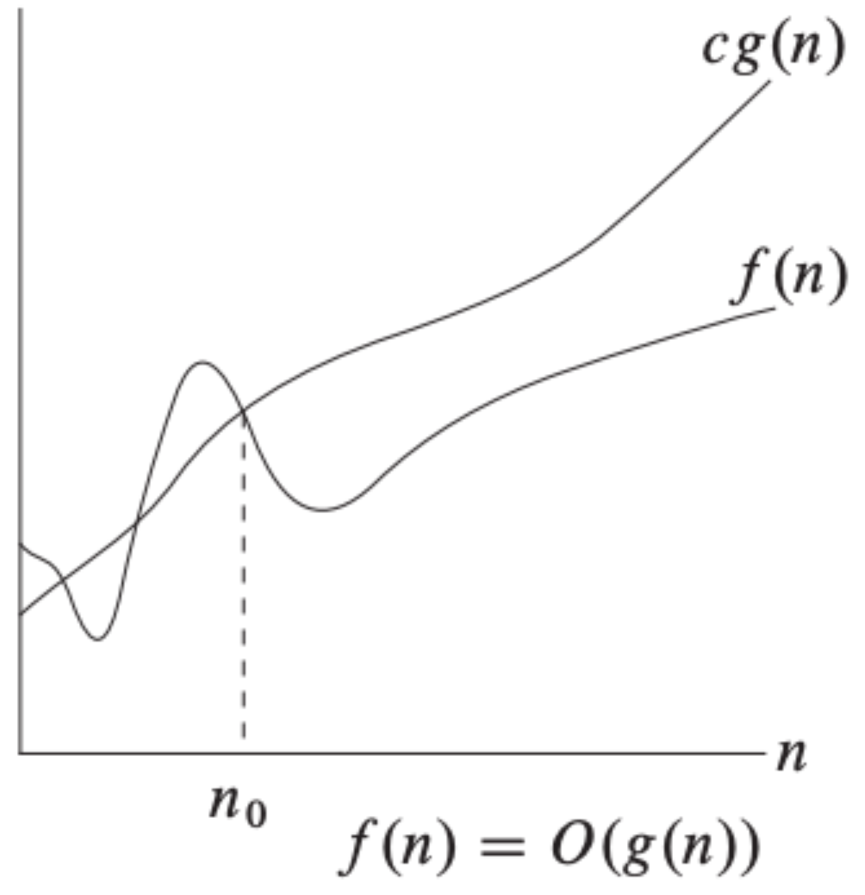
Algorithm analysis

- Correctness
  - Pre-condition and post-conditions for each procedure (esp. recursive calls).
  - Loop **invariant** for iterative algorithms.
  - Termination in finte steps. E.x. decreasing of non-negative measure.
- Resource analysis
  - Recurrences. $T(n) = 2T(n/2) + 3n$. Recursion tree, Master theorem.
  - Amortized analysis, probabilistic analysis, ...
  - Experimentation.
- Model of computation
  - E.x. random-access machine (RAM). Unit cost per instruction and memory access.

Asymptotic notations

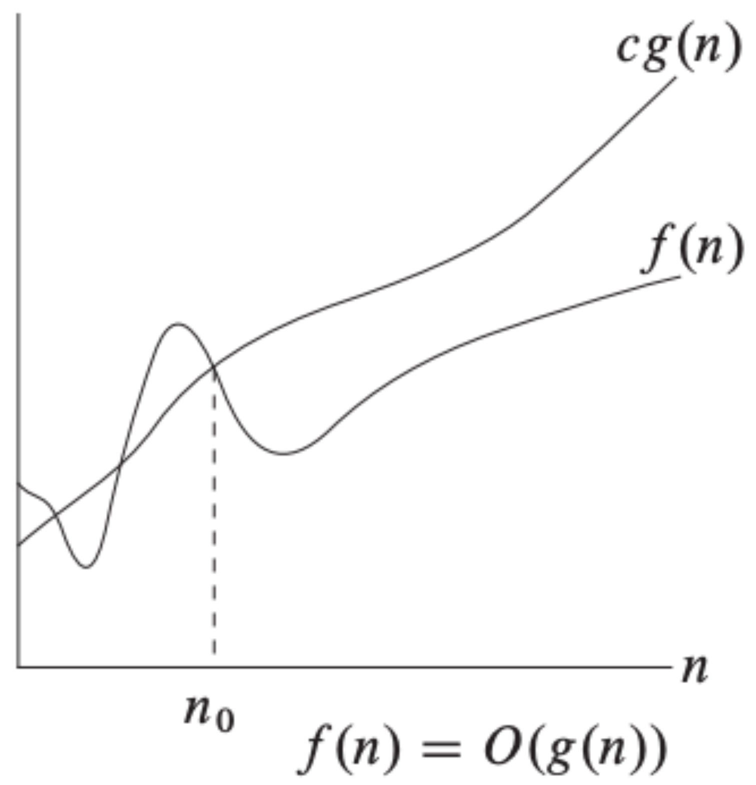
- $O(\cdot), \Omega(\cdot), \Theta(\cdot), o(\cdot), \omega(\cdot)$
  - Measure algorithm behaviors (by functions on integers) as problem size grows.
  - Usually a good indicator of which alg. is preferable (except for small inputs).
- **Defining $O(\cdot)$: asymptotic upper bound**

We write $f(n) = O(g(n))$ if there exist constants $c > 0, n_0 > 0$, such that $0 \leq f(n) \leq cg(n)$ for all $n \geq n_0$.



- **$O(g(n))$ as a set**

$$O(g(n)) := \{f(n) : \exists c > 0, n_0 > 0, \text{ such that } 0 \geq f(n) \leq c \cdot g(n) \text{ for all } n \geq n_0\}$$



Examples

$f(n) = O(g(n))$ if there exist constants $c > 0, n_0 > 0$, such that $0 \leq f(n) \leq cg(n)$ for all $n \geq n_0$.

- $2n^2 = O(n^3)$
  - $c = 1, n_0 = 2.$
  - I.e., $2n^2 \in O(n^3)$
- $f(n) = n^3 + O(n^2)$
  - Meaning $f(n) = n^3 + h(n)$ for some $h(n) \in O(n^2)$

Exercise: sort by asymptotic order of growth

1. $n \log n$
2. $\sqrt{n}$
3. $\log n$
4. n^2
5. 2^n
6. n
7. $n!$
8. $n^{1,000,000}$
9. $n^{1/\log n}$
10. $\log(n!)$

List them in ascending order: if f appears before g, then $f = O(g)$

9, 3, 2, 6, 1=10, 4, 8, 5, 7

→

Summary

Notation	… means …	Think…	E.g.	Lim $f(n)/g(n)$
$f(n)=O(n)$	$\exists\ c>0, n_0>0, \forall\ n > n_0$: $0 \leq f(n) < cg(n)$	Upper bound	$100n^2 = O(n^3)$	If it exists, it is $< \infty$
$f(n)=\Omega(g(n))$	$\exists\ c>0, n_0>0, \forall\ n > n_0$: $0 \leq cg(n) < f(n)$	Lower bound	$n^{100} = \Omega(2^n)$	If it exists, it is > 0
$f(n)=\Theta(g(n))$	both of the above: $f=\Omega(g)$ **and** $f = O(g)$	Tight bound	$\log(n!) = \Theta(n \log n)$	If it exists, it is > 0 and $< \infty$
$f(n)=o(g(n))$	$\forall\ c>0, n_0>0, \forall\ n > n_0$: $0 \leq f(n) < cg(n)$	Strict upper bound	$n^2 = o(2^n)$	Limit exists, $=0$
$f(n)=\omega(g(n))$	$\forall\ c>0, n_0>0, \forall\ n > n_0$: $0 \leq cg(n) < f(n)$	Strict lower bound	$n^2 = \omega(\log n)$	Limit exists, $=\infty$

LaTeX

- Start with Overleaf.com
- Questions? Turn to Google and tex.stackexchange.com
- Short math guide for LaTeX

- Getting more serious?
  - A good text editor and packages/plugins, e.g., Emacs + AUCTeX
  - Version-control: GitHub

★ These things take time to learn, but you will not regret the effort!

★ Scrutinize your PDFs to see if anything could be nicer-looking.

★ Take pity on your poor reader.

LaTeX Live Demo

- **What, why?**
- **Homework template file**
  - Preamble,
  - Basic formatting: italics, bold, lists, reference, comments, table, etc.
  - Soln environment
- **Math**
  - Good practices

Scratch