https://algs4.cs.princeton.edu

Algorithm Design

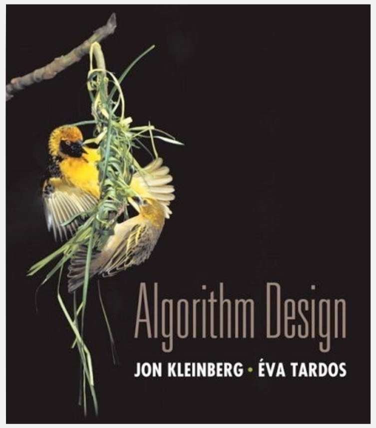
- *analysis of algorithms*
- *greedy*
- *network flow*
- *dynamic programming*
- *divide-and-conquer*
- *randomized algorithms*

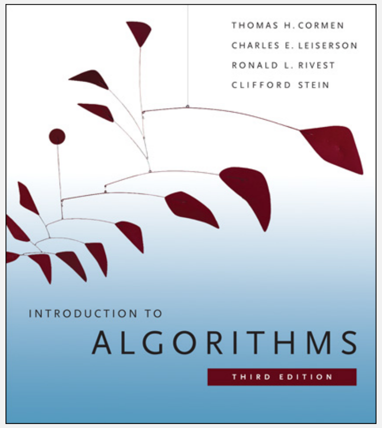
Last updated on 12/12/19 9:43 AM

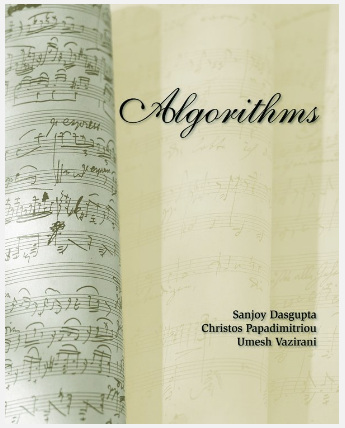
Algorithm design

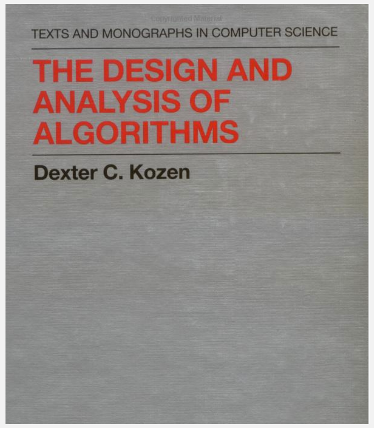
Algorithm design patterns.

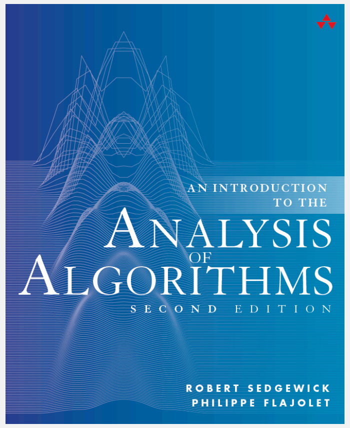
- Analysis of algorithms.
- Greedy.
- Network flow.
- Dynamic programming.
- Divide-and-conquer.
- Randomized algorithms.











Want more? See COS 340, COS 343, COS 423, COS 445, COS 451, COS 488, ...

Interview questions

https://algs4.cs.princeton.edu

Algorithm Design

- *analysis of algorithms*
- *greedy*
- *network flow*
- *dynamic programming*
- *divide-and-conquer*
- *randomized algorithms*

Egg drop

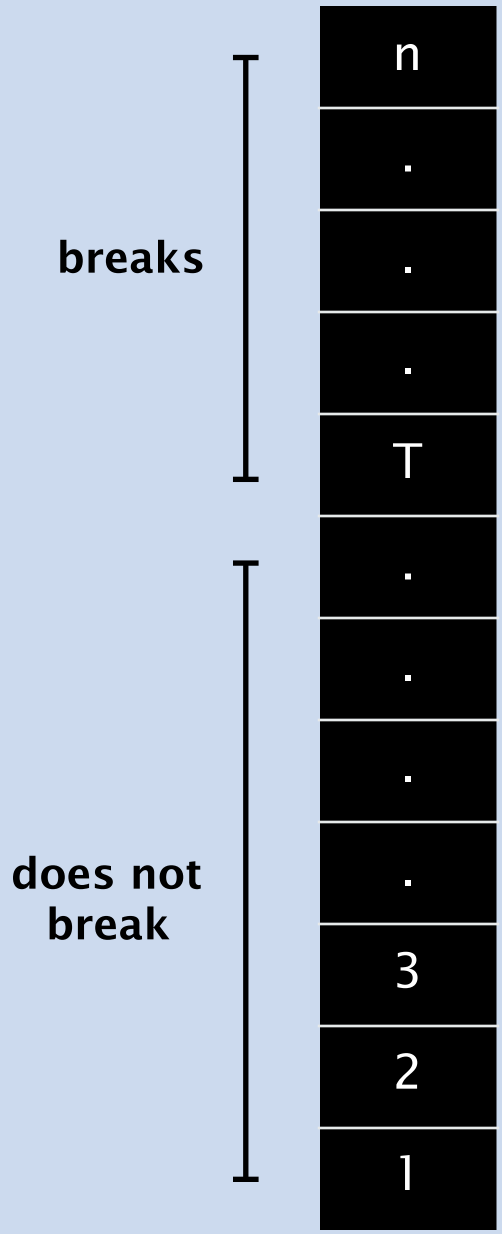
Goal. Find T using fewest number of tosses.

Variant 0. 1 egg.

Variant 1. ∞ eggs.

Variant 2. ∞ eggs and $\sim 2 \lg T$ tosses.

Variant 3. 2 eggs.



https://algs4.cs.princeton.edu

Algorithm Design

- *analysis of algorithms*
- ***greedy***
- *network flow*
- *dynamic programming*
- *divide-and-conquer*
- *randomized algorithms*

Greedy algorithms

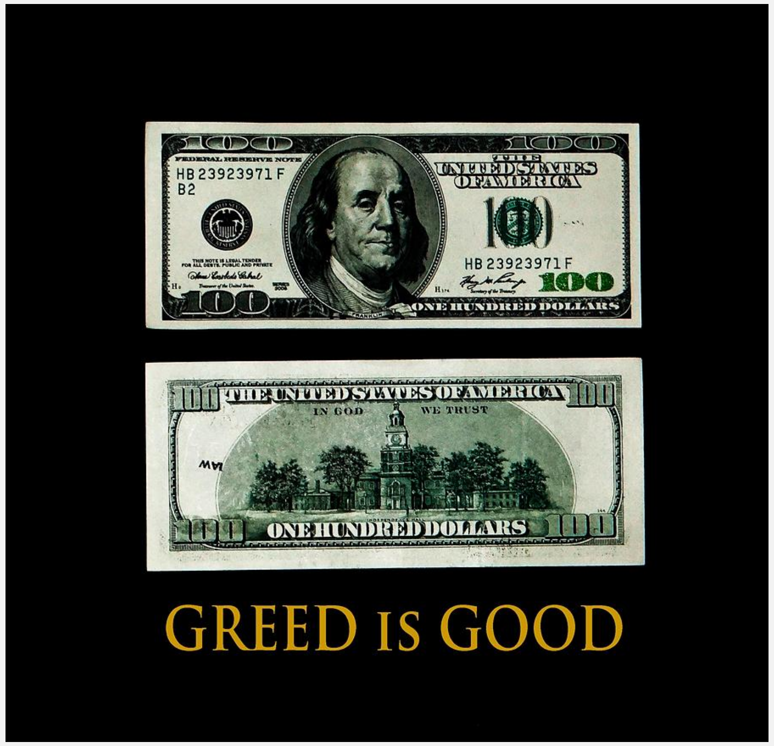
Make locally optimal choices at each step.

Familiar examples.

- Huffman coding.
- Prim's algorithm.
- Kruskal's algorithm.
- Dijkstra's algorithm.

More classic examples.

- U.S. coin changing.
- Activity scheduling.
- Gale–Shapley stable marriage.
- ...



Caveat. Greedy algorithm rarely leads to globally optimal solution.
(but is often used anyway, especially for intractable problems)

Document search

Given a document that is a sequence of n words, and a query that is a sequence of m words, find the smallest range in the document that includes the m query words (in the same order).

Ex. Query = "textbook programming computer"

```
This book is intended to survey the most important computer
algorithms in use today, and to teach fundamental techniques
to the growing number of people in need of knowing them. It
is intended for use as a textbook for a second course in
computer science, after students have acquired basic
programming skills and familiarity with computer systems.
The book also may be useful for self-study or as a reference
for people engaged in the development of computer systems or
applications programs, since it contains implementations of
useful algorithms and detailed information on performance
characteristics and clients.
```

https://algs4.cs.princeton.edu

Algorithm Design

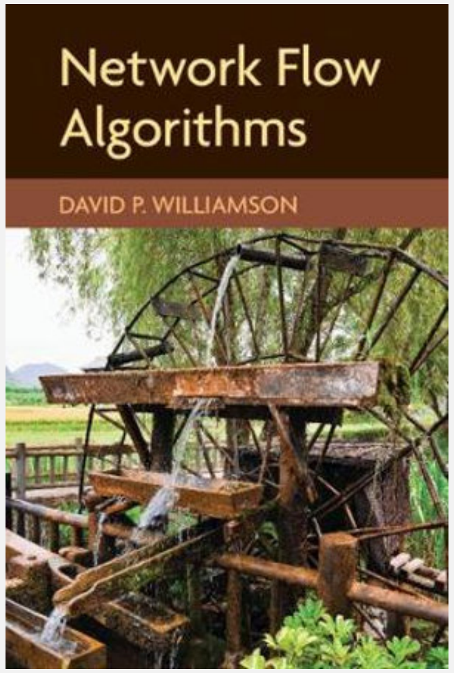
- analysis of algorithms
- greedy
- network flow
- dynamic programming
- divide-and-conquer
- randomized algorithms

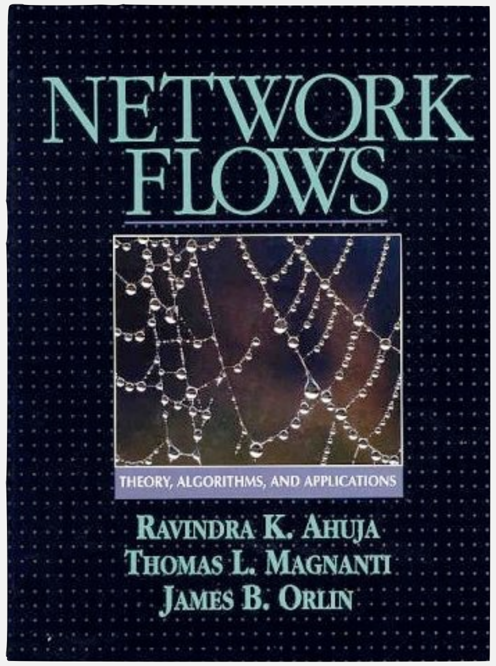
Network flow

Classic problems on edge-weighted graphs.

Familiar examples.

- Shortest paths.
- Bipartite matching.
- Maxflow and mincut.
- Minimum spanning tree.





Other classic examples.

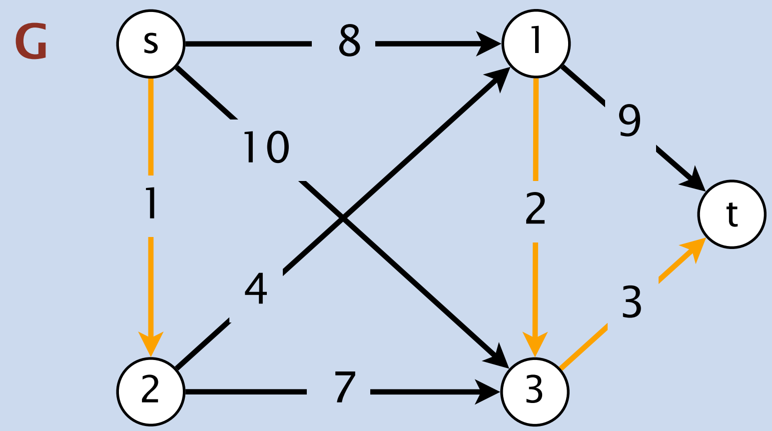
- Minimum-cost arborescence.
- Non-bipartite matching.
- Assignment problem.
- Minimum-cost flow.
- ...

"reduction"

Applications. Many many problems can be modeled using network flow.

Shortest path with orange and black edges

Goal. Given a digraph, where each edge has a positive weight and is orange or black, find shortest path from s to t that uses at most k orange edges.



k = 0: s→1→t (17)

k = 1: s→3→t (13)

k = 2: s→2→3→t (11)

k = 3: s→2→1→3→t (10)

https://algs4.cs.princeton.edu

Algorithm Design

- *analysis of algorithms*
- *greedy*
- *network flow*
- *dynamic programming*
- *divide-and-conquer*
- *randomized algorithms*

Dynamic programming

- Break up problem into a series of overlapping subproblems.
- Build up solutions to larger and larger subproblems.
  (caching solutions to subproblems in a table for later reuse)

Familiar examples.

- Shortest paths in DAGs.
- Seam carving.
- Bellman–Ford.

THE THEORY OF DYNAMIC PROGRAMMING

RICHARD BELLMAN

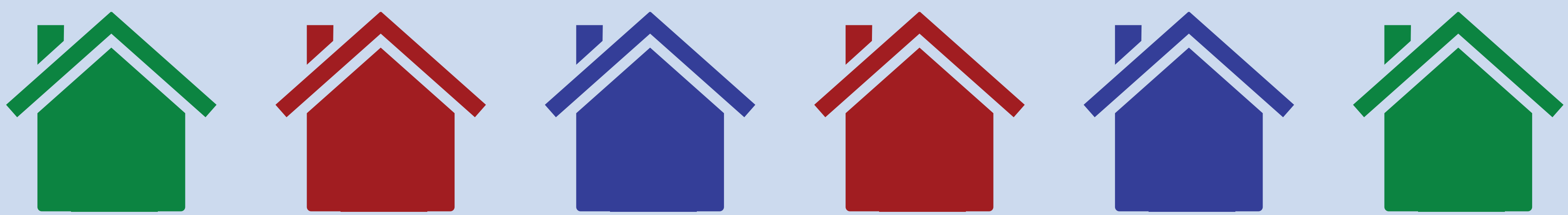
More classic examples.

- Unix diff.
- Viterbi algorithm for hidden Markov models.
- Smith–Waterman for DNA sequence alignment.
- CKY algorithm for parsing context-free grammars.
- ...

House coloring problem

Goal. Paint a row of n houses red, green, or blue so that

- No two adjacent houses have the same color.
- Minimize total cost, where $cost(i, color)$ is cost to paint i given color.



	A	B	C	D	E	F
	7	6	7	8	9	20
	3	8	9	22	12	8
	16	10	4	2	5	7

cost to paint house i the given color
(3 + 6 + 4 + 8 + 5 + 8 = 34)

https://algs4.cs.princeton.edu

Algorithm Design

- analysis of algorithms
- greedy
- network flow
- dynamic programming
- *divide-and-conquer*
- randomized algorithms

Divide and conquer

- Break up problem into two or more independent subproblems.
- Solve each subproblem recursively.
- Combine solutions to subproblems to form solution to original problem.

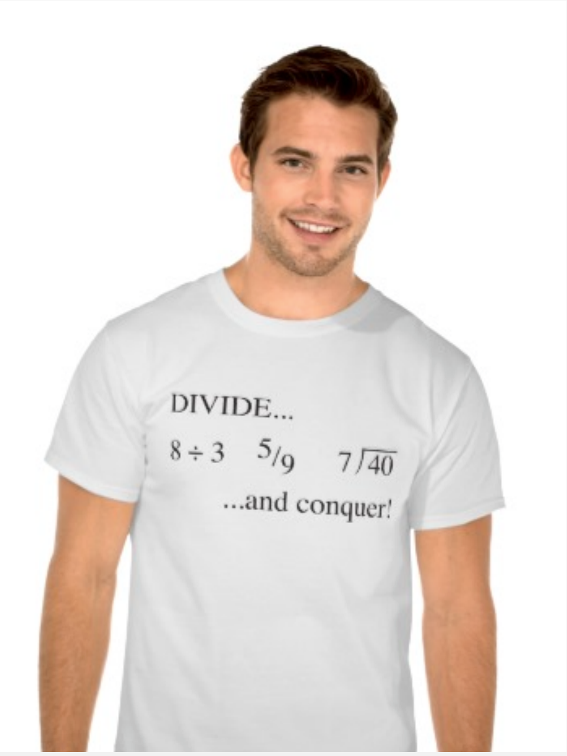
Familiar examples.

- Mergesort.
- Quicksort.

More classic examples.

- Closest pair.
- Convolution and FFT.
- Matrix multiplication.
- Integer multiplication.

...



needs to take COS 226?

Prototypical usage. Turn brute-force n^2 algorithm into $n \log n$ algorithm.

https://algs4.cs.princeton.edu

Algorithm Design

- analysis of algorithms
- greedy
- network flow
- dynamic programming
- divide-and-conquer
- randomized algorithms

Randomized algorithms

Algorithm that uses random coin flips to guide its behavior.

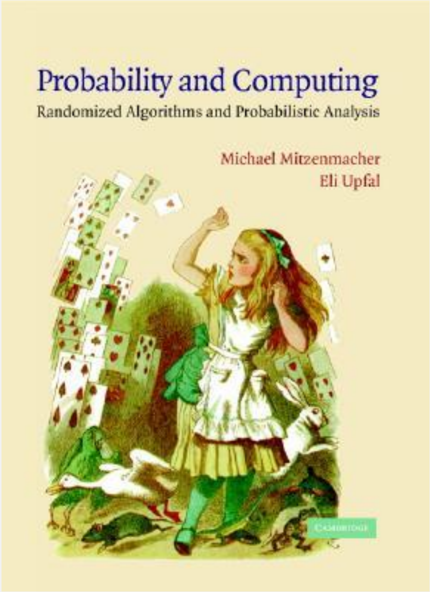
Familiar examples.

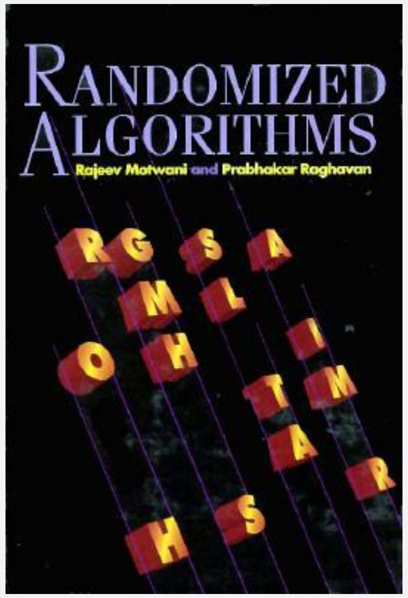
- Quicksort.
- Quickselect.

More classic examples.

- Rabin–Karp substring search.
- Miller–Rabin primality testing.
- Polynomial identity testing.
- Volume of convex body.
- Universal hashing.
- Global min cut.

...





Nuts and bolts

Problem. A disorganized carpenter has a mixed pile of n nuts and n bolts.

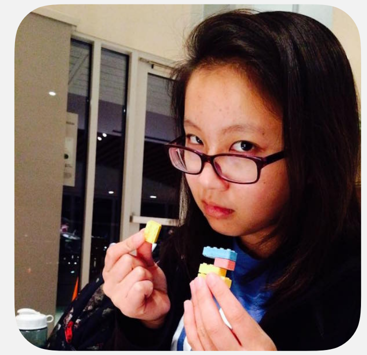
- The goal is to find the corresponding pairs of nuts and bolts.
- Each nut fits exactly one bolt and each bolt fits exactly one nut.
- By fitting a nut and a bolt together, the carpenter can see which one is bigger (but cannot directly compare either two nuts or two bolts).

Brute-force n^2 solution. Compare each bolt to each nut.

Challenge. Design an $n \log n$ algorithm.

Credits

Faculty lead preceptors, Turing preceptor, and graduate student AIs.

Undergraduate graders and lab TAs. Apply to be one next semester!

Ed tech. Several developed here at Princeton!

IJ iClicker piazza CS Princeton Salon

gradescope codePost Quizzera

A farewell video (from P04, Fall 2018)

" Algorithms and data structures are love.

Algorithms and data structures are life. "

— anonymous COS 226 student