Algorithm Design

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

https://algs4.cs.princeton.edu
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
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. $\infty$ eggs.
Variant 2. $\infty$ eggs and $\sim 2 \lg T$ tosses.
Variant 3. 2 eggs.
Algorithm Design

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

https://algs4.cs.princeton.edu
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)
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”
Algorithm Design

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

https://algs4.cs.princeton.edu
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.
- ...

**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.

![Graph G with orange and black edges](image)

- $k = 0$: $s \rightarrow 1 \rightarrow t$ (17)
- $k = 1$: $s \rightarrow 3 \rightarrow t$ (13)
- $k = 2$: $s \rightarrow 2 \rightarrow 3 \rightarrow t$ (11)
- $k = 3$: $s \rightarrow 2 \rightarrow 1 \rightarrow 3 \rightarrow t$ (10)
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.

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.

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>7</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>20</td>
</tr>
<tr>
<td>B</td>
<td>3</td>
<td>8</td>
<td>9</td>
<td>22</td>
<td>12</td>
<td>8</td>
</tr>
<tr>
<td>C</td>
<td>16</td>
<td>10</td>
<td>4</td>
<td>2</td>
<td>5</td>
<td>7</td>
</tr>
</tbody>
</table>

Cost to paint house $i$ the given color

$(3 + 6 + 4 + 8 + 5 + 8 = 34)$
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.
  ...

Prototypical usage. Turn brute-force $n^2$ algorithm into $n \log n$ algorithm.
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.
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!
A farewell video (from P04, Fall 2018)

COS 226 P04 Presents..
“Algorithms and data structures are love.
Algorithms and data structures are life.”
— anonymous COS 226 student