CS10: The Beauty and Joy of Computing

Lecture #6: Algorithms

Photo credit to genericitis
http://www.flickr.com/photos/genericitis/

In the news:

A recently security study suggests that a computer worm that ran rampant several years ago is still running on many machines, including 50% of Fortune 500 companies and US government agencies.

The worm is capable of redirecting the user's web requests and installing arbitrary software.

What is the world speed record for solving a 3x3x3 Rubik's cube?

a) 12 minutes, 3 seconds

b) 58.1 seconds

c) 7.96 seconds

d) 5.66 seconds

e) 3.31 seconds

Rubik's Cube Champion

Feliks Zemdegs

http://www.youtube.com/watch?v=3v_Km6cv6DU

What is an algorithm?

An algorithm is any well-defined computational procedure that takes some value or set of values as input and produces some value or set of values as output.

The *concept* of algorithms, however, is far older than computers.

Early Algorithms

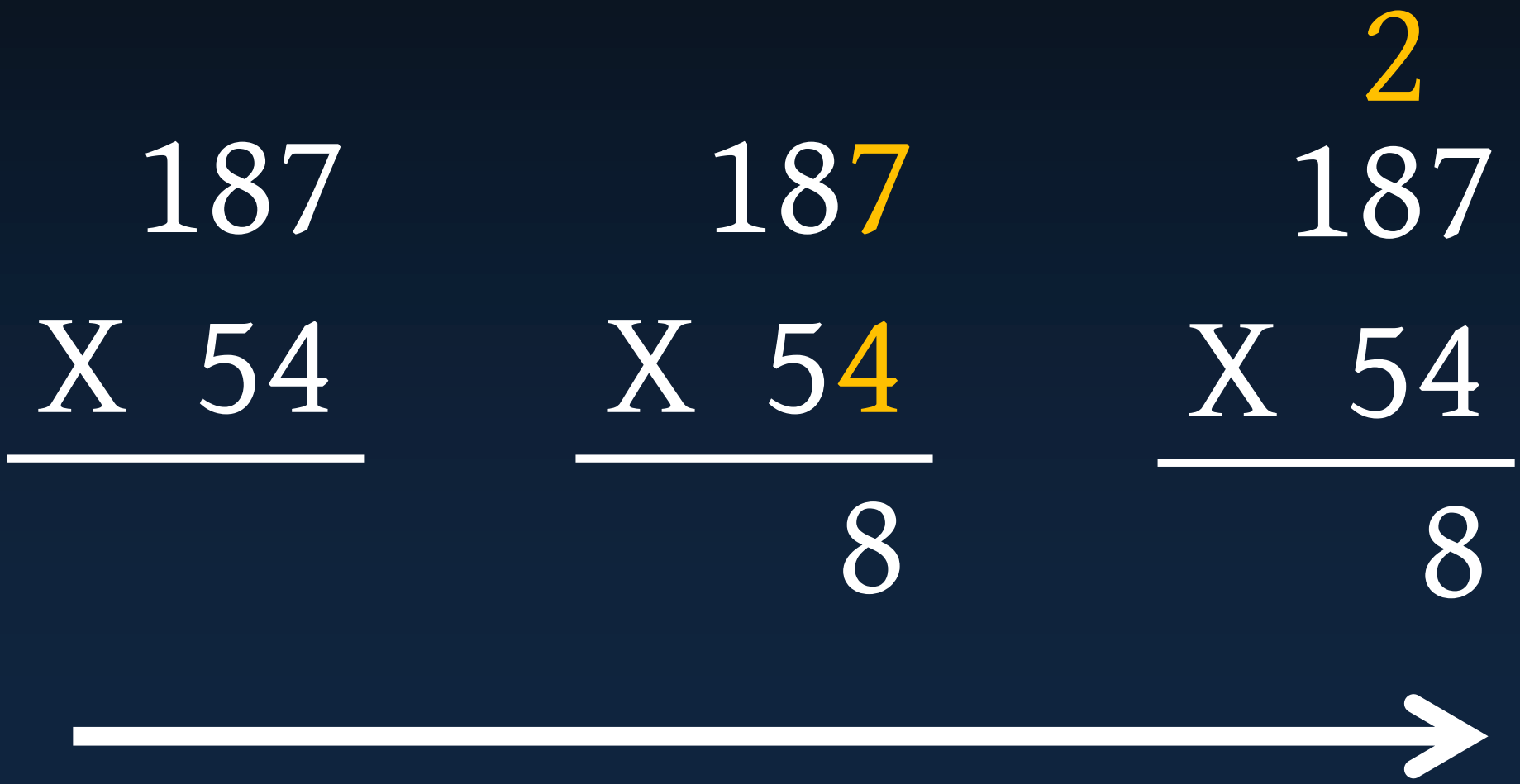
Photo credit to Daniel Niles

Dances, ceremonies, recipes, and building instructions are all conceptually similar to algorithms.

Babylonians defined some fundamental mathematical procedures ~3,600 years ago.

Algorithms You've Seen

Multiplication algorithm (for humans)



Algorithms You've Seen

Length of word

Whether a word appears in a list

Whether a list is sorted

Pick a random word of length *x* from list

Commonly Used Algorithms

Luhn algorithm
Credit card number validation

Damerau–Levenshtein distance
Spell-checkers

PageRank
Google's way of measuring "reputation" of web pages

EdgeRank
Facebook's method for determining what appears in your news feed

Choosing a Technique

Most problems can be solved in more than one way, meaning that those problems have multiple algorithms to describe how to find the solution.

Not all of these algorithms are created equal. Very often we have to make some trade-offs when we select a particular one.

We'll talk more about this next time.

Ways to Attack Problems

There are many different categories of algorithms. Three common groups are:

"Brute force"

Keep trying stuff until something works.

Top-down

Divide the full problem up into smaller subproblems.

Bottom-up

Start with simple solutions and build up to complex ones.

Algorithms vs. Functions & Procedures

A function or procedure is an implementation of an algorithm.

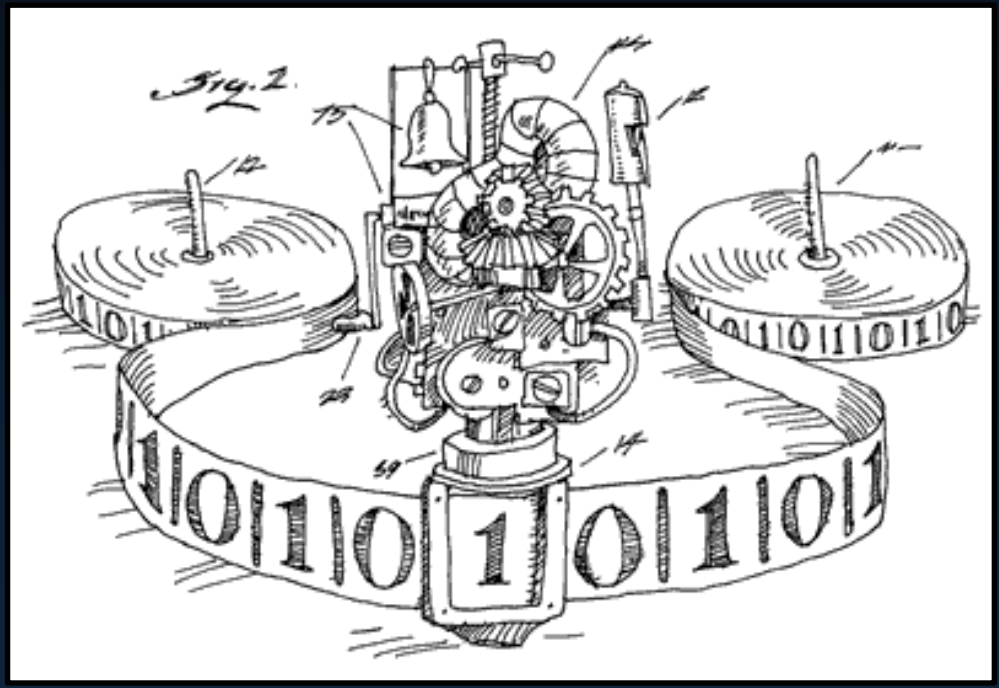
Algorithms are conceptual definitions of how to accomplish a task and are language agnostic. Functions and procedures are written in particular languages and can be run to produce results.

Turing Completeness

A language that is Turing complete can do anything that other Turing complete languages can do.

A language cannot be more powerful than a Turing complete language.

What does it mean for a language to be Turing complete?



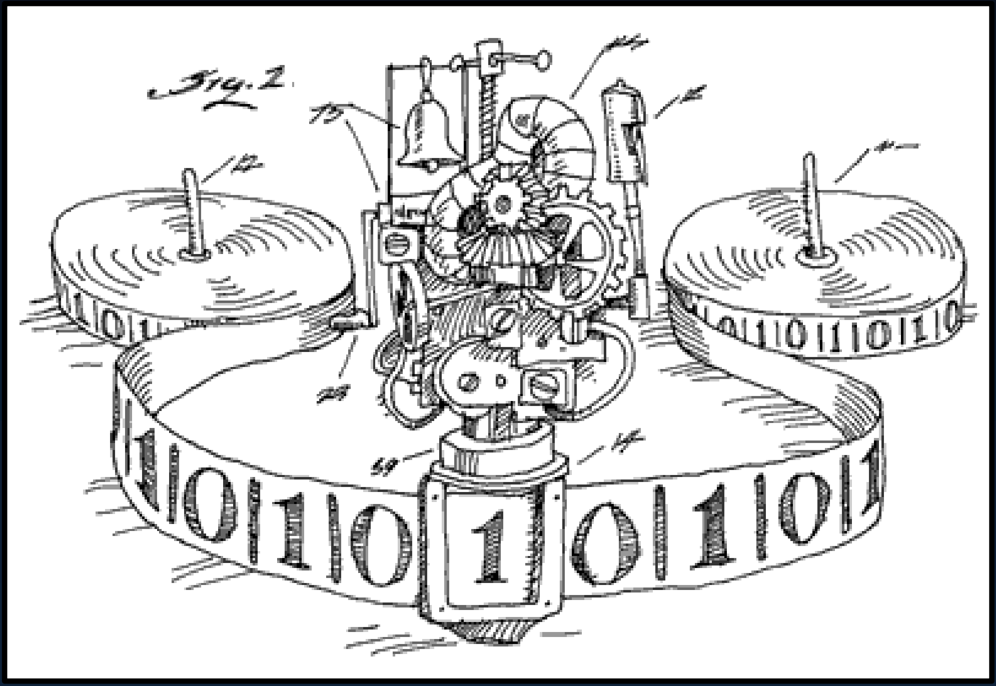
Turing Machine by Tom Dunne

Turing Completeness

A Turing machine is a conceptual device made up of an infinitely long tape (think VCR's) and a device that can read & write data to the tape.

A universal Turing machine is a Turing machine that can simulate any Turing machine.

A Turing complete language is the real-world equivalent of a UTM.



Turing Machine by Tom Dunne

Algorithm Correctness

We don't only want algorithms to be fast and efficient; we want them to be correct!

TOTAL Correctness

Always reports, and the answer is always correct.

PARTIAL Correctness

Sometimes reports, and the answer is always correct *when it reports.*

We also have probabilistic algorithms that have a certain *probability* of returning the right answer.

Summary

- The concept of an algorithm has been around forever, and is an integral topic in CS.
- Algorithms are well-defined procedures that can take inputs and produce output.
- We're constantly dealing with trade-offs when selecting / building algorithms.
- Correctness is particularly important and testing is the most practical strategy to ensure this.