CS10: The Beauty and Joy of Computing
Lecture #6: Algorithms

In the news:

Two researchers have discovered how to crack the SSL suite of security algorithms that underlie the majority of the web's secure connections. They're currently reporting the ability to hijack connections to secure websites in < 10 minutes.

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

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

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

Algorithms You've Seen

Multiplication algorithm (for humans)

$$\begin{array}{r} 187 \\ \text{X}\ \ 54 \\ \hline \\ \end{array} \qquad \begin{array}{r} 187 \\ \text{X}\ \ 54 \\ \hline 8 \end{array} \qquad \begin{array}{r} 2 \\ 187 \\ \text{X}\ \ 54 \\ \hline 8 \end{array}$$

$$\longrightarrow$$

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

WPA
Wireless security

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.

Algorithm Correctness

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

TOTAL Correctness	PARTIAL Correctness
Always reports, and the answer is always correct.	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.

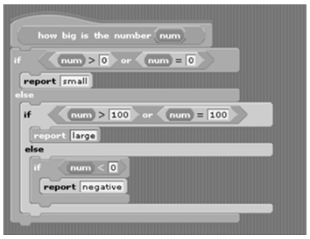
Correctness is Important

Most widely-used algorithms have been mathematically proven to be correct over all inputs in their domains.

This often isn't practical in every case, so we fall back on an alternative method instead: *testing!*

If there are problems, users will find them!

Checking Correctness



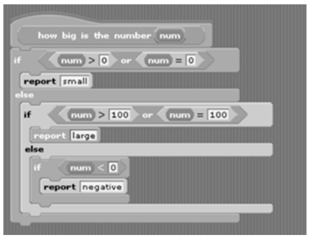
```
how big is the number (num)
if <<(num) > [0]> or <(num) = [0]>>
  report [small]
else
  if <<(num) > [100]> or <(num) = [100]>>
    report [large]
  else
    if <(num) < [0]>
      report [negative]
```

* note: this block has bugs

Important to test normal cases as well as edge cases.

Cover as many paths of execution as possible.

Checking Correctness



```
how big is the number (num)
if <<(num) > [0]> or <(num) = [0]>>
  report [small]
else
  if <<(num) > [100]> or <(num) = [100]>>
    report [large]
  else
    if <(num) < [0]>
      report [negative]
```

* note: this block has bugs

Which set of inputs would provide the best test cases for this block?

a) 1, 10, 20, 100

b) -10, 0, 50, 900

c) -90, -80, 0, 150

d) -1, 0, 1, 2

e) no testing required... I'm positive that it works.

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.