CSC104 winter 2013

Why and how of computing
week 2

Danny Heap
heap@cs.toronto.edu
BA4270 (behind elevators)
http://www.cdf.toronto.edu/~heap/104/W13/
416-978-5899

Text: Picturing Programs

Computer Science
UNIVERSITY OF TORONTO

could algorithms run the world?

Spectacular algorithm success leads to questions:

- Is there, potentially, an algorithm to solve every problem?

  is there an app?

- If there are two or more algorithms solving the same problem, how do you choose?

  we'll see some.

- How do you discover new algorithms?

  things that *might* work

problems without an algorithm

before electronic, programmable computers
Alonzo Church and Alan Turing showed there were many unsolvable problems

(H P ← I)

Classic example: Halting Problem

another example

If there an algorithm for each problem, how about one to decide whether declarative English sentences are true? How about:

$T(S)$

$S =$ This statement is false.

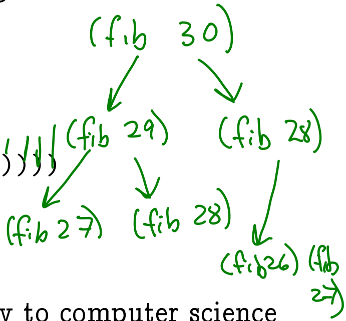
What should the algorithm that verifies (or not) sentences do?

algorithms that take too long

An algorithm may exist, but take too long to be feasible:

```
(define (fib n)
  (if (< n 2)
      n
      (+ (fib (- n 1)) (fib (- n 2)))))
```

opaque.



Of interest from rabbit-breeding to biology to computer science (see Vi Hart), calculating Fibonacci sequence **this way** gets slow for numbers over 40.

an everyday (once) algorithm

Before on-line dictionaries, it was common to look up definitions in a paper-and-ink dictionary. There are (at least) two different, correct ways to find the leaf (2-sided sheet) with the word you're looking for (or conclude it's not in the dictionary).

- linear search
- binary search

how to solve it

it being a new problem

Clearly there's no fool-proof method, but there's some techniques that often make progress. It helps to write down the whole process:

- Understand the problem
- Devise (one or more) plan(s)
- Try the plan
- Look back

paper folding?

try it out

- Understand the problem (what's given, what's required)?
- Devise a plan
- Try at least one plan (be ready to abandon it too)
- Look back

Notes