UC Berkeley EECS
Lecturer
Gerald Friedland

# The Beauty and Joy of Computing

## Lecture #6
## Algorithms I

**Quest** (first exam) in **in 2 days!!**

University students 'made to wear anti-cheating helmets'

Students in Thailand appear to have been forced to wear helmets to prevent them from cheating during exams.

In an effort to curb rampant cheating, Bangkok's Kasetsart University created an anti-deceit paper helmet for students to wear during their midterm exams. Photo: Facebook

## 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 fundamental mathematical procedures ~3,600 years ago.
- Genes contain algorithms!

Photo credit: Daniel Niles

## Algorithms You've Seen in CS10

- Length of word
- Whether a word appears in a list
- Interact with the user (ask)
- Word Comparisons (You wrote one for HW1!)
- Sort a List (see lab!)
- Make this a block!

## Algorithms You Might Have Heard Of

**Luhn algorithm**
Credit card number validation

**Deflate**
Lossless data compression

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

**EdgeRank**
Facebook's method for determining what is highest up on your news feed

## Important Terms

**Sequencing**
Application of each step of an algorithm in order (sometimes: find order)

**Selection**
Use of Boolean condition to select execution parst

**Iteration**
Repetition of part of an algorithm until a condition is met

**Recursion**
Repeated application of the same part of algorithm on smaller problems

## Properties of Algorithms

- Algorithm + Algorithm = Algorithm
- Part of Algorithm = Algorithm
- Algorithms can be efficient or inefficient given a comparison algorithm
- Several algorithms may solve the same problem

## Algorithm Correctness

We don't only want algorithms to be fast and efficient; we also 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.

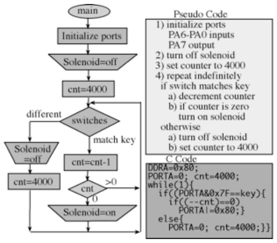
## How to Express Algorithms…

A programmer's spouse tells him: "Run to the store and pick up a loaf of bread. If they have eggs, get a dozen." The programmer comes home with 12 loaves of bread.

Algorithms need to be expressed in a context-free, unambiguous way for all participants

## Ways to Express Algorithms

- Natural Language
- Pseudo Code
- Programming Language



Pseudo Code

```
1) initialize ports
     PA6-PA0 inputs
     PA7 output
2) turn off solenoid
3) set counter to 4000
4) repeat indefinitely
   if switch matches key
      a) decrement counter
      b) if counter is zero
           turn on solenoid
   otherwise
      a) turn off solenoid
      b) set counter to 4000
```

C Code

```
DDRA=0x80;
PORTA=0; cnt=4000;
while(1){
  if((PORTA&0x7F==key){
    if((--cnt)==0)
      PORTA|=0x80;}
  else{
    PORTA=0; cnt=4000;}}
```

- …or in any other information conveying way!

## Programming Languages

**C/C++**
Good for programming that is close to hardware

**Java/C#**
Portable code

**Python/Perl/TclTK**
Fast to write and portable

**BASIC/BYOB/SNAP**
Good for teaching programming concepts

All programming languages can be used to implement (almost) any algorithm!

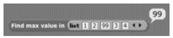
## Choosing a Technique

- Most problems can be solved in more than one way, i.e., multiple algorithms exist to describe how to find the solution.
- The right language makes formulating algorithms easier and clearer
- Not all of these algorithms are created equal. Very often we have to make some trade-offs when we select a particular one.
- There are unsolvable problems!

## Algorithms vs. Functions & Procedures

- Algorithms are conceptual definitions of how to accomplish a task and are language agnostic, usually written in pseudo-code.
- Find max value in list
  - Set (a temporary variable) the max as the first element
  - Go through every element, compare to max, and if it's bigger, replace the max
  - Return the max

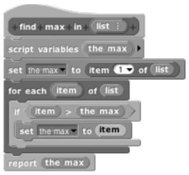
- A function or procedure is an implementation of an algorithm, in a particular language.



```
Find max value in list 1 2 99 3 4
```

99

- Find max value in list



```
find max in list
script variables the max
set the max to item 1 of list
for each item of list
    if item > the max
        set the max to item
report the max
```

## 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. Programming languages help us express them.
- We're constantly dealing with trade-offs when selecting / building algorithms.
- Correctness is particularly important and testing is the most practical strategy to ensure it.
  - Many write tests first!