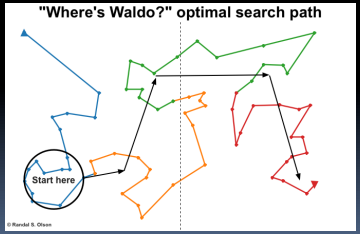
UC Berkeley EECS
Sr Lecturer SOE
Dan Garcia

The Beauty and Joy of Computing

Lecture #6
Algorithms

Optimal Algorithm For Finding Waldo…

A researcher used a genetic algorithm – one that "evolves" over different generations, and competes against a metric of success (here, distance the eye has to travel) to find the optimal search path for finding Waldo.



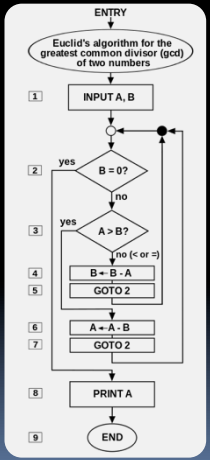
www.randalolson.com/2015/02/03/heres-waldo-computing-the-optimal-search-strategy-for-finding-waldo/

Algorithms: Definitions

Algorithm: Definition

- "Algorithms are precise sequences of instructions for processes that can be executed by a computer and are implemented using programming languages."
- The concept of algorithms, however, is far older than computers.

Euclid's GCD Algorithm
(*Wikipedia,* Somepics)



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!

Woman Basket Weaving
(*Wikipedia*, Public Domain)

Algorithms You've Seen in BJC so far

- 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!)

Algorithms You May Already Know

Luhn algorithm
Credit card number validation

Deflate
Lossless data compression

PageRank
Google's way to measure web page "reputation"

EdgeRank
Facebook's way to determine news feed sort

Building Blocks of Algorithms

Sequencing

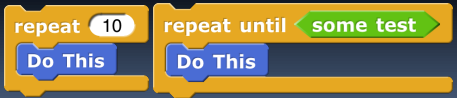
Application of each step of an algorithm in order given

Selection

Use of Boolean condition to select which of two parts to do

Iteration

Repetition algorithm part # times or until condition met



Recursion

The overall algorithm calls itself to help solve the problem on smaller parts,combine result. (we'll see later)

Every algorithm can be constructed using only Sequencing, Selection, & Iteration!

(Cal) Which of the following is false?

a) Algorithms can be worth *billions* of $

b) Paul Revere practiced *selection*

c) You learned your first algorithm before you could speak

d) Proving algorithms are *correct* is easy

e) Algorithms can *adapt,* like a living thing

Algorithms: Properties, Expressing

Properties of Algorithms

- Algorithms can be combined to make new algorithms.
- Using existing correct algorithms as building blocks for constructing a new algorithm helps ensure the new algorithm is correct.
- Knowledge of standard algorithms can help in constructing new algorithms
- Different algorithms can be developed to solve the same problem.
- Developing a new algorithm to solve a problem can yield insight into the problem

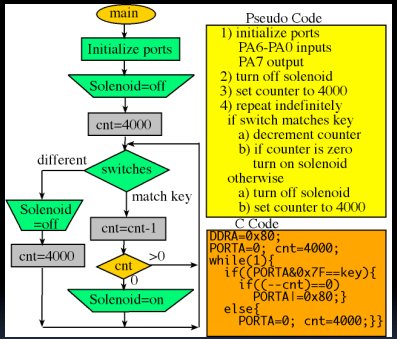
How to Express Algorithms…

A programmer's spouse says: "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!

Languages for Algorithms

- Natural Language, Pseudo Code
  - For Humans to understand
- Visual & Text-based Programming Languages
  - Can be run on a computer
- …or in any other information conveying way!



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

Which Language to Choose?

- Different languages are better suited for expressing different algorithms
- Some programming languages are designed for specific domains and are better for expressing algorithms in those domains
- The language used to express an algorithm can affect characteristics such as clarity or readability but not whether an algorithmic solution exists
- Clarity and readability are important considerations when expressing an algorithm in a language.

Programming Languages

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

Java/C#
Portable code

Python/Perl/TclTK
Fast to write and portable

Scratch/Snap!
Good for teaching programming concepts

Nearly all programming languages are equivalent in terms of being able to express any algorithm!

(Cal) Of 4 paradigms, what's the *most* powerful?

a) Functional

b) Imperative

c) OOP

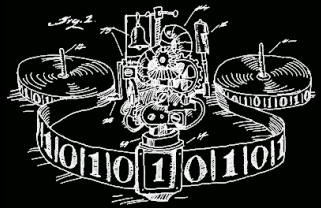
d) Declarative

e) All equally powerful

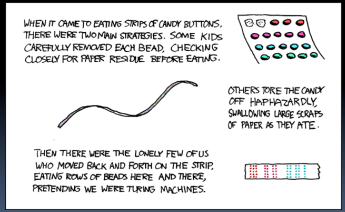
Algorithms: Turing Completeness

Turing Completeness

- A Turing Machine has an infinite tape of 1s and 0s and instructions that say whether to move the tape left, right, read, or write it
  - Can simulate any computer algorithm!
- A Universal Turing Machine is one that can simulate a Turing machine on any input
- A language is considered Turing Complete if it can simulate a Universal Turing Machine
  - A way to decide that one programming language or paradigm is just as powerful as another



Turing Machine by Tom Dunne



Xkcd comic "Candy Button Paper"

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.
- Each paradigm / language has its unique benefits
  - All Turing complete languages are equally powerful
  - Paradigms vary in efficiency, scalability, overhead, fun, "how" vs "what", …