Algorithms

CSE 120 Spring 2017

Instructor: Justin Hsia

Teaching Assistants: Anupam Gupta, Braydon Hall, Eugene Oh, Savanna Yee

Administrivia

- Assignments:
  - Jumping Monster due Saturday (4/15)
  - Mice and Predator due Sunday (4/16)
  - Creativity Planning due Tuesday (4/18)
    - Find a partner, come up with *two* proposed programs

Outline

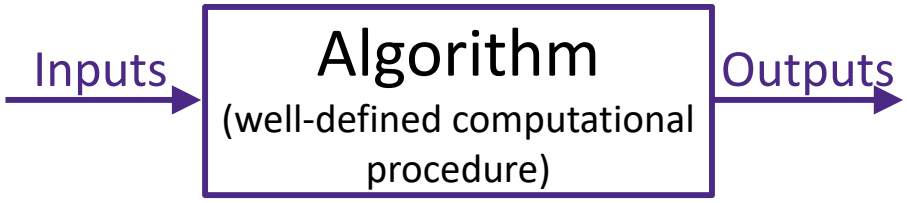
- **Algorithms**
- Examples of Algorithms
- Specifying Algorithms

Definition

- 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."

  "An algorithm is thus a *sequence of computational steps* that transform the input into the output."

  - From textbook *Introduction to Algorithms* (link)



Computational Problems

- Can think of an algorithm as a tool for solving a computational problem
  - The problem statement specifies desired input/output (I/O) relationship
  - The algorithm describes a specific computational *procedure* that gives you the desired input/output (I/O) relationship

- Example: Sorting is a computational problem
  - Problem statement: Given a sequence of numbers, put them in order
  - Example I/O: [1, 9, 3, 6, 2] → [1, 2, 3, 6, 9]

Early Algorithms

- The concept of algorithms pre-dates computers
  - Dances, ceremonies, recipes, and building instructions are all *conceptually similar* to algorithms
  - Mathematical algorithms go way back:
    - Babylonians defined many fundamental procedures ~3600 years ago, more formal algorithms in Ancient Greece
    - Al-Khwarizmi laid out many algorithms for computation using decimal numbers
    - You implicitly use hundreds of numerical algorithms!
  - Nature runs algorithms (*e.g.* genes and development)

Properties of Algorithms

- Algorithms can be combined to make new algorithms
  - It helps to know standard algorithms
  - Building from correct algorithms helps ensure correctness
- There are many algorithms to solve the same computational problem
- Developing a new algorithm to solve a problem can lead to insight about the problem
  - Example: 3SUM – Given a list of n real numbers, are there three numbers in the list that sum to zero? New algorithm in 2014 broke supposed speed barrier

Outline

- Algorithms
- **Examples of Algorithms**
- Specifying Algorithms

Algorithms You've Seen

- Draw a square using lines
- Make a character move into place on the screen
- Make a multi-colored grid of animals
- … and many more!

An Algorithm You've Seen Before

- Multiplying two numbers:

```
  2
 23
x 7
---
161
```

- Another multiplication algorithm?
  - Common core "box" method:

	20	3
7	140	21

140+21=161

Algorithm Demo Time

- I need 10 brave(?) volunteers to join me at the front of the class!
  - 7 of you are numbers (grab a sheet of paper)
  - 3 of you will represent algorithms

1) Find the Smallest Number in a List

- Input: 6 numbers in a given order
- Output: The *index* of the smallest number

- Example: {6, 3, 5, 1, 2, 4} would output 4, since the smallest number is in the 4^{th} position

- Algorithm:

2) Search an Unordered List

- Input: 6 numbers in a given order, desired number
- Output: Yes/True if desired number is in the list, No/False otherwise

- Algorithm:

3) Sort an Unordered List

- Input: 6 numbers in a given order
- Output: The same list, but now in numerical order

- Example: {6, 3, 5, 1, 2, 4} would output {1, 2, 3, 4, 5, 6}

- Algorithm:

4) Find Median of a List

- **Note:** This is an actual job interview question!
- Problem: Given a list of numbers (odd length, no repeats), return the median
- Example: {9, 2, 1, 6, 3, 4, 7} would output 4
- Algorithm:

End of Demo

- Round of applause for our volunteers!

More Famous Algorithms

- PageRank algorithm
  - Google's measure of the "reputation" of web pages
- EdgeRank algorithm
  - Facebook's method for determining what to show on your News Feed
- Luhn algorithm
  - Credit card number validation
- Deflate
  - Lossless data compression
- RSA Encryption
  - Encrypt (secure) data for transmission

Outline

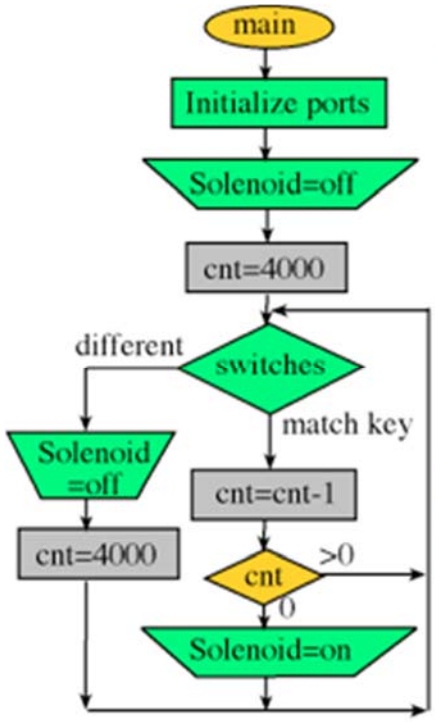
- Algorithms
- Examples of Algorithms
- **Specifying Algorithms**

Be Specific!

- A programmer's spouse says, "Run to the store and pick up a loaf of break. If there have eggs, get a dozen."
- What happens?
  - The programmer comes home with 12 loaves of bread!
- Algorithms need to be expressed in an *unambiguous* way for all participants

Ways to Express Algorithms

- Many ways to specify an algorithm:
  - Natural Language (*e.g.* English, 中國) or Pseudocode
    - Easy for humans to understand, but can be ambiguous or vague
  - Visual and text-based programming languages (*e.g.* Scratch, Processing)
    - Have an exact meaning with no ambiguity
    - Can be run on a computer
  - Other information-conveying ways
    - *e.g.* flowcharts or pictures



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;}}
```

Google Query

- From Larry Page and Sergey Brin's research paper *The Anatomy of a Large-Scale Hypertextual Web Search Engine*

1. Parse the query.
2. Convert words into wordIDs.
3. Seek to the start of the doclist in the short barrel for every word.
4. Scan through the doclists until there is a document that matches all the search terms.
5. Compute the rank of that document for the query.
6. If we are in the short barrels and at the end of any doclist, seek to the start of the doclist in the full barrel for every word and go to step 4.
7. If we are not at the end of any doclist go to step 4.
   Sort the documents that have matched by rank and return the top k.

Figure 4. Google Query Evaluation

Implementations

- If we specify an algorithm using code in a programming language, we say that the code implements the algorithm
  - A function or program that can be run on a computer

- Example: Find index of smallest in list
  - Algorithm 1 of demo
    - Pseudocode/natural language
  - Function in Processing
    - Implementation

```
// returns the index of the smallest number in a list
int find_smallest(float[] list) {
  int smallest = 0;
  for(int i = 1; i < list.length; i=i+1) {
    if(list[i] < list[smallest]) {
      smallest = i;
    }
  }
  return smallest;
}
```

Which Language to Use?

- Different languages are better suited for expressing different algorithms
  - Some languages are designed for specific domains, and are better for expressing algorithms in those domains
    - e.g. Unix shell scripts, SQL, HTML
  - Language choice can affect things like efficiency, portability, clarity, or readability
    - Clarity and readability are VERY important considerations
    - Doesn't affect existence of algorithmic solution

Programming Languages

- Programming language sweet spots:
  - **C/C++** Code that is close to the hardware
  - **Java/C#** Portable code
  - **Python** Fast to write
  - **Javascript** Great for running in web browsers
  - **Processing** Great with visuals and interaction

- Most programming languages can implement (almost) ANY algorithm
  - Equally “powerful”

Peer Instruction Question

- I hand you a cooking recipe with no name – is this equivalent to a *computational problem*, *algorithm*, or *implementation*?
  - What can and can't I do with just the recipe?
  - Vote at http://PollEv.com/justinh

A. Computational Problem

B. Algorithm

C. Implementation

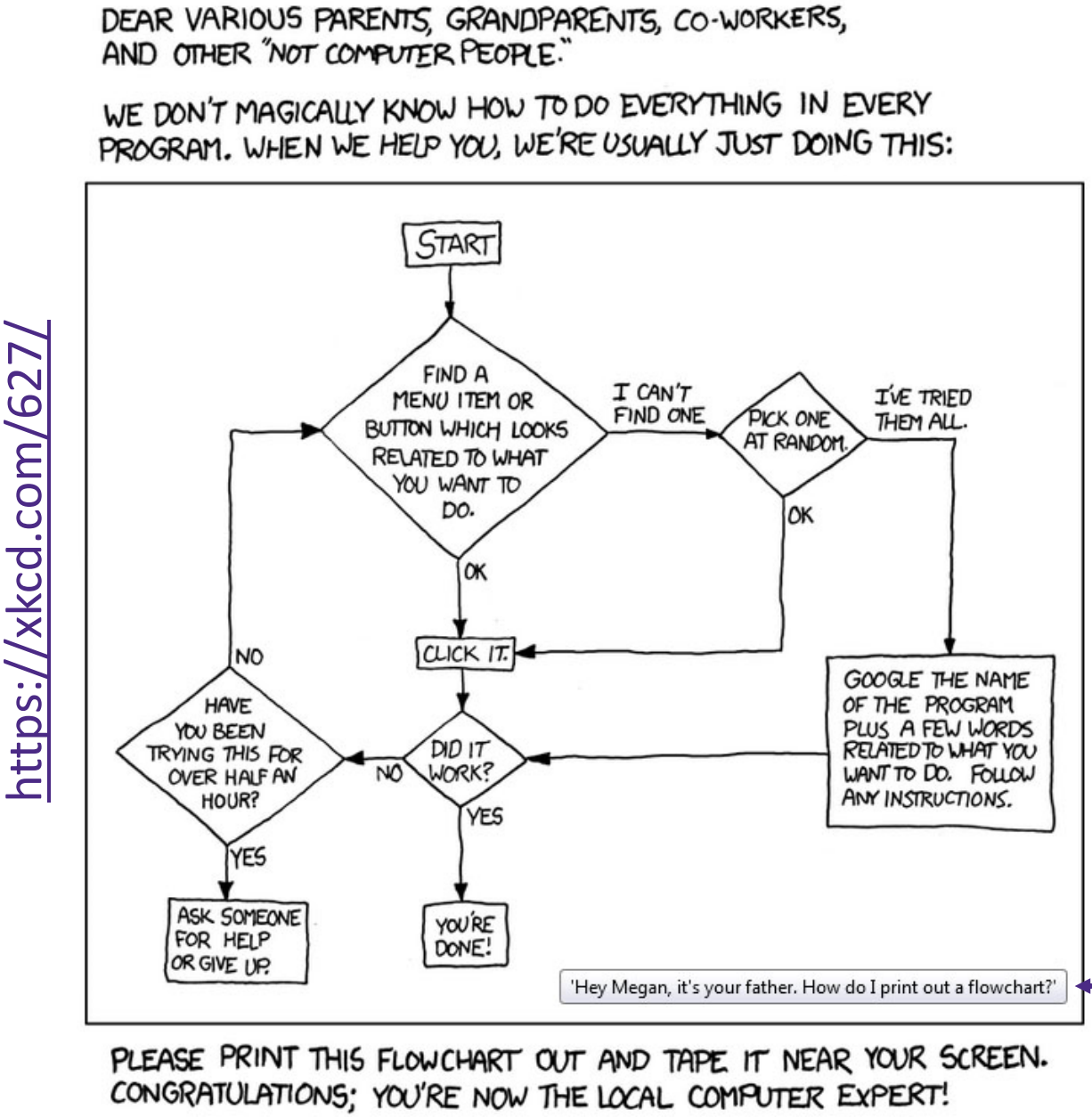
Summary

- A computational problem is a problem statement with desired I/O relationship
- An algorithm *describes* a computational procedure that satisfies/solves a computational problem
- An implementation is an algorithm written out in a programming language (unambiguous, executable)

- Properties of algorithms:
  - Can be combined to make new algorithms
    - Knowledge of existing algorithms & correctness help
  - There are many algorithms to solve the same computational problem

Summary

https://xkcd.com/627/



← "alt" text