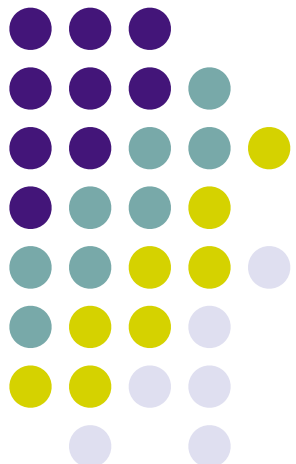
Algorithms
Part 1

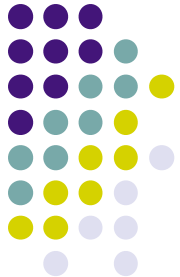
CMSC 104, Spring 2014

Christopher S. Marron

(thanks to John Park for slides)



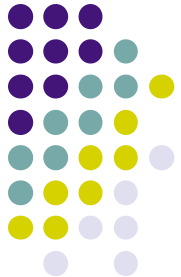
Algorithms, Part 1



Topics

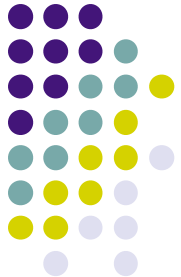
- Definition of an Algorithm
- Algorithm Examples
- Syntax versus Semantics

Problem Solving



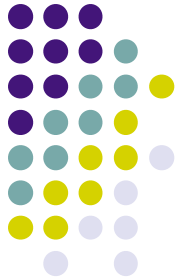
- Problem solving is the process of transforming the description of a problem into the solution of that problem.
- We use our knowledge of the problem domain (requirements).
- We rely on our ability to select and use appropriate problem-solving strategies, techniques, and tools.

Algorithms



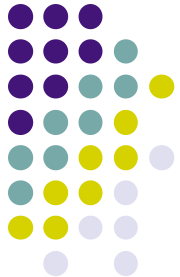
- An **algorithm** is a step by step solution to a problem.
- Why bother writing an algorithm?
  - For your own use in the future. You won't have to rethink the problem.
  - So others can use it, even if they know very little about the principles behind how the solution was derived.

Examples of Algorithms



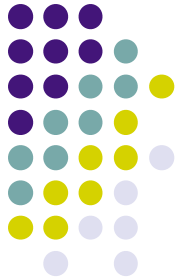
- Washing machine instructions
- Instructions for a ready-to-assemble piece of furniture
- Two classics:
  - Finding the greatest common divisor (GCD) using Euclid's Algorithm
  - Sorting the elements in a list of data, e.g. sorting a list of numbers

Washing Machine Instructions



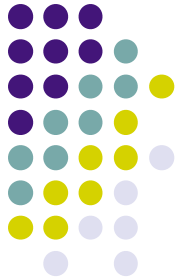
- Separate clothes into white clothes and colored clothes.
- Add 1 cup of powdered laundry detergent to tub.
- For white clothes:
  - Set water temperature knob to HOT.
  - Place white laundry in tub.
- For colored clothes:
  - Set water temperature knob to COLD.
  - Place colored laundry in tub.
- Close lid and press the start button.

Observations About the Washing Machine Instructions



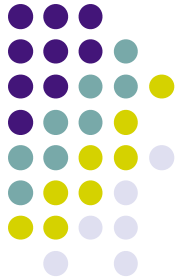
- There are a finite number of steps.
- We are capable of doing each of the instructions.
- When we have followed all of the steps, the washing machine will wash the clothes and then will stop.

Refinement of Algorithm Definition



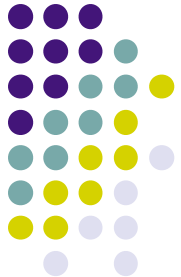
- Our old definition:
  - An algorithm is a step by step solution to a problem.
- Adding our observations:
  - An algorithm is a finite set of executable instructions that directs a terminating activity.

Instructions for a Ready-to-Assemble Piece of Furniture



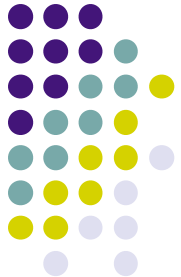
- “Align the marks on side A with the grooves on Part F.”
- How could these instructions be hard to follow?
  - Which side is A? A & B look alike -- both line up with Part F! This instruction is ambiguous.

Final Version of the Algorithm Definition



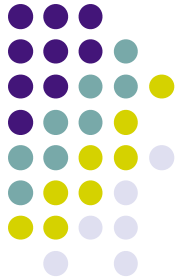
- Our old definition:
  - An algorithm is a finite set of executable instructions that directs a terminating activity.
- Final version:
  - An algorithm is a finite set of unambiguous, executable instructions that directs a terminating activity.

History of Algorithms



- The study of algorithms began as a subject in mathematics.
- The search for algorithms was a significant activity of early mathematicians.
- Efficiency was important - there were no computers!
- Goal: find a single set of instructions that can be used to solve any problem of a particular type efficiently (a **general solution**).

Euclid's Algorithm

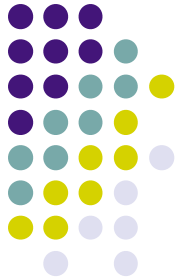


Problem: Find the largest positive integer that divides evenly into two given positive integers (i.e., the **greatest common divisor**).

Algorithm:

1. Assign M and N the values of the larger and smaller of the two positive integers, respectively.
2. Divide M by N and call the remainder R.
3. If R is not 0, then assign M the value of N, assign N the value of R, and return to Step 2. Otherwise, the greatest common divisor is the value currently assigned to N.

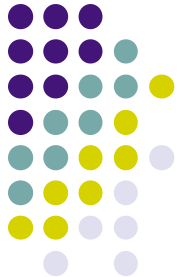
Finding the GCD of 24 and 9



M	N	R
24	9	6
9	6	3
6	3	0

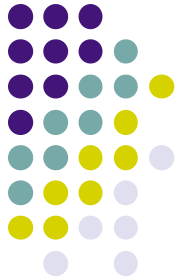
So, 3 is the GCD of 24 and 9.

Euclid's Algorithm (con't)



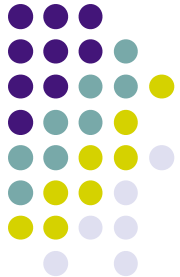
- Do we need to know the theory that Euclid used to come up with this algorithm in order to use it?
- What intelligence is required to find the GCD using this algorithm?

The Idea Behind Algorithms



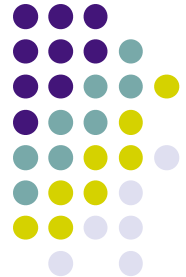
- Once an algorithm behind a task has been discovered – we can just use it!
  - We don't need to understand the principles.
  - The task is reduced to following the instructions.
  - The intelligence is "encoded into the algorithm."

Algorithm Representation



- Syntax and Semantics
  - **Syntax** refers to the representation itself.
  - **Semantics** refers to the concept represented (i.e., the logic).

Contrasting Syntax and Semantics

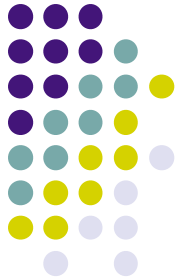


- In the English language, we have both syntax and semantics.
- Syntax is the grammar of the language.
- Semantics is the meaning.
- Given the following sentence,

  I walked to the corner grocery store.

  - Is this sentence syntactically correct?
  - Is it semantically correct?

Contrasting Syntax and Semantics (con't)



- Given the following sentence,

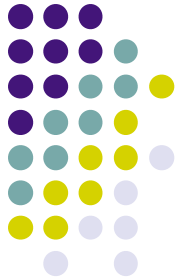
  I talked to the funny grocery store.

  - Is this sentence syntactically correct?
  - Is it semantically correct?

- How about

  I grocery store walked corner the to.

Contrasting Syntax and Semantics (con't)



- Conclusion: An English sentence may be syntactically correct, yet semantically incorrect.
- This is also true of algorithms.
- And it is also true of computer code.