
Algorithms

INFO/CSE 100, Autumn 2004
Fluency in Information Technology

<http://www.cs.washington.edu/100>

Readings and References

- Reading
 - » *Fluency with Information Technology*
 - Chapter 10, Algorithmic Thinking

Algorithm

- Algorithm
 - » a precise, systematic method to produce a desired result
- For example, the placeholder technique for deleting a short string except where it occurs in longer strings is an algorithm with an easy specification:

```
longStringWithShortStringInIt ← placeholder
ShortString ← ε
placeholder ← longStringWithShortStringInIt
```

Properties of Algorithms

- For an algorithm to be well specified it must have ...
 - » Inputs specified
 - The range of possible inputs is well defined
 - » Outputs specified
 - The desired output is well defined
 - » Definiteness
 - The steps to take are definite and understandable
 - » Effectiveness
 - The steps must be possible to accomplish
 - » Finiteness
 - A processor that follows the algorithm will eventually finish

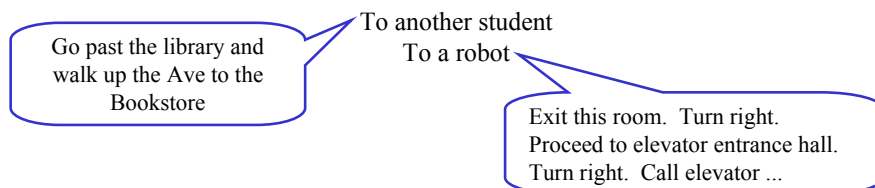
Communicating with People

- People can fill in missing steps, but can get swamped by lots of details and clutter
- What helps when communicating with people?
 - » Be *organized*, stay on the main point
 - » Invent *abstractions* to help convey the basic ideas quickly
 - » *Interact* with the other person to make sure that you are “on the same page”

Communicating with Computers

- Computers cannot fill in missing steps, but can manage lots and lots of detail without error
- What helps when communicating with computers?
 - » Be *organized* and consistent in all the details
 - » Invent *abstractions* to help specify the basic ideas accurately and consistently
 - » *Analyze* your algorithm and its implementation, because you won't get to interact later

Example: Directions to the Bookstore



- The student operates at a higher level of *abstraction* with a richer *vocabulary* of *shorthands*
- An *algorithm* is a plan for how to accomplish a task
 - » A *program* is an implementation of an algorithm
- Good algorithms (at any level of abstraction) require *precision*

Algorithms Analysis: What is it?

- What is an algorithm?
 - » A sequence of steps that accomplishes a task
- Many different algorithms may correctly solve a given task
 - » can it be implemented with available equipment?
 - » will it complete within this lifetime?
 - » will it require gigabytes of memory?

Algorithm Analysis: Why do it?

- Understand the mathematical fundamentals needed to analyze algorithms
- Learn how to compare the efficiency of different algorithms in terms of running time and memory usage
- Study a number of standard algorithms for data manipulation and learn to use them for solving new problems

Multiplication Algorithm

What is 112×8 ?

What is 112×18 ?

Programs vs Algorithms

- A program is an algorithm specialized to a particular situation

» an Algorithm

`longStringWithShortStringInIt ← placeholder`

`ShortString ← ε`

`placeholder ← longStringWithShortStringInIt`

» a Program that implements the Algorithm

`↵↵ ← # // replace double <newlines> with <#>`

`↵ ← ε // delete all single <newlines>`

`# ← ↵↵ // restore all double <newlines>`

Programming as Communication

- When we write a program, we are communicating with
 - » the computer
 - » other people
- The computer reads our program as the set of instructions that it should perform
 - » It just needs to know how, not why
- Other people read our programs to understand how and why
 - » Programs that don't work (bugs)
 - » Program evolution - new features
 - » Performance improvement

An algorithm to alphabetize CDs

```
define variable named Artist
  use Artist to refer to the name of the group that made a CD
for all slots in the rack starting at one end
  call the current slot alpha
  for all the remaining slots in the rack
    call the next slot beta
    Exchange?
      If Artist of the CD in the beta slot is earlier in the
      alphabet than the Artist of the CD in the alpha slot,
      interchange the CDs
    next beta
  next alpha
done
```

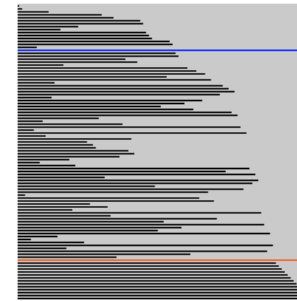
22-Oct-2004

cse100-10-algorithms © 2004 University of Washington

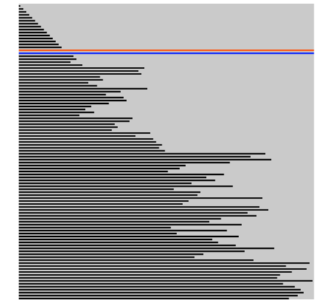
13

Sort Demo

Bubble Sort



Quick Sort



<http://java.sun.com/applets/jdk/1.0/demo/SortDemo/example1.html>

22-Oct-2004

cse100-10-algorithms © 2004 University of Washington

14

Another version

```
public static void sort(String[] list) {
  for (int alpha=0; alpha<list.length; alpha++) {
    for (int beta=alpha+1; beta<list.length; beta++) {
      if (list[beta].compareTo(list[alpha]) < 0) {
        String temp = list[alpha];
        list[alpha] = list[beta];
        list[beta] = temp;
      }
    }
  }
}
```

```
> java Sorter cat bird ant dog elephant fox
--Unsorted--
cat bird ant dog elephant fox
--Sorted--
ant bird cat dog elephant fox
>
```

Sorter.java and drJava

22-Oct-2004

cse100-10-algorithms © 2004 University of Washington

15

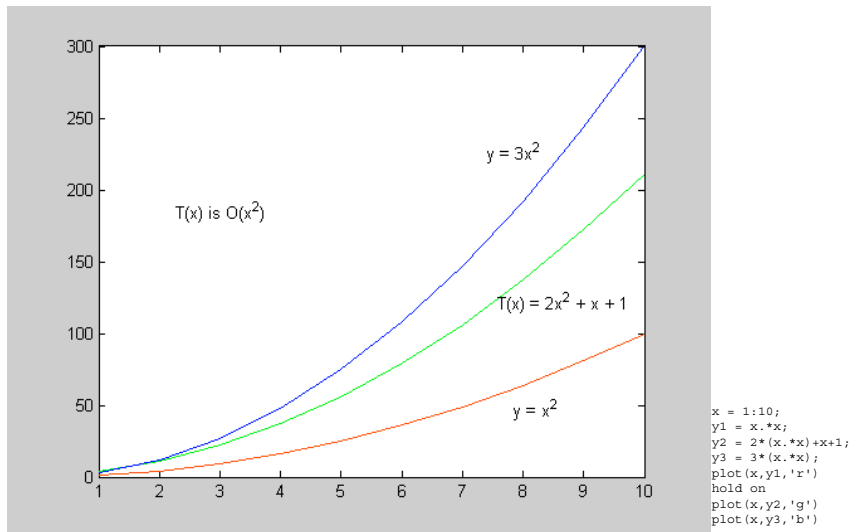
Metaphor: Programs as Math

- We also can think of programs as *executable math*:
a program calculates some result for us.
- Consider:
$$Area = \pi \cdot Radius^2$$
- We can employ such expressions in programs.
- Most of our intuitions and knowledge about mathematics apply to computers.

22-Oct-2004

cse100-10-algorithms © 2004 University of Washington

16



Using the program Matlab to calculate and plot function values

Summary

- We can figure out many algorithms on our own, abstracting from specific cases
- We can learn from others who have studied particular algorithms in depth
- We abstract parts of an algorithm or program to understand them
 - » Thinking of how the program works and reasoning about its properties allows us to know why an algorithm works ... and then we can get the computer to do it for us