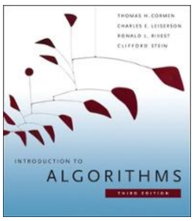
Introduction to Algorithms



Chapter 1: The Role of Algorithms in Computing

Computational Problems

- A computational problem specifies an input-output relationship
  - What does the input look like?
  - What should the output be for each input?
- Example:
  - Input: an integer number n
  - Output: Is the number prime?
- Example:
  - Input: A list of names of people
  - Output: The same list sorted alphabetically

Algorithms

- A tool for solving a well-specified computational problem

Input → $\boxed{\textbf{\textit{Algorithm}}}$ → ***Output***

- Algorithms must be:
  - Correct: For each input produce an appropriate output
  - Efficient: run as quickly as possible, and use as little memory as possible – more about this later

Algorithms

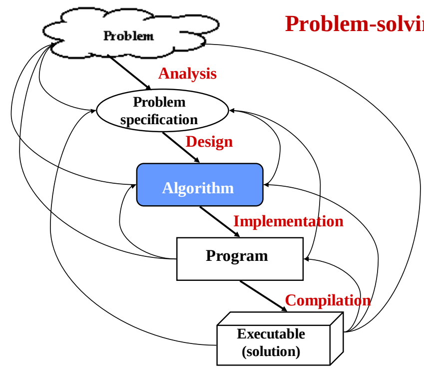
- A well-defined computational procedure that takes some value, or set of values, as input and produces some value, or set of values, as output.
- A method of solving a problem, using a sequence of well-defined steps.
- Written in a pseudo code which can be implemented in the language of programmer's choice.

Problems and Algorithms

- We need to solve a computational problem
  - "Convert a weight in pounds to Kg"
- An algorithm specifies how to solve it, e.g.:
  - 1. Read weight-in-pounds
  - 2. Calculate weight-in-Kg = weight-in-pounds * 0.455
  - 3. Print weight-in-Kg
- A computer program is a computer-executable description of an algorithm

L1.5

Problem-solving Process



L1.6

The problem of sorting

Input: sequence $\langle a_1, a_2, \ldots, a_n \rangle$ of numbers.

Output: permutation $\langle a'_1, a'_2, \ldots, a'_n \rangle$ such that $a'_1 \leq a'_2 \leq \cdots \leq a'_n$.

Example:

Input: 8 2 4 9 3 6

Output: 2 3 4 6 8 9

L1.7

Instances of a problem

- An algorithm is said to be ***correct*** if for every input instance, it halts with the correct output
- An ***instance of a problem*** consists of all inputs needed to compute a solution to the problem
- A correct algorithm ***solves*** the given computational problem. An incorrect algorithm might not halt at all on some input instance, or it might halt with other than the desired answer

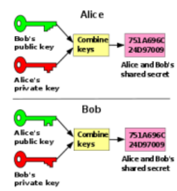
L1.8

What kind of problem are solved by algorithms? (1/2)

- The Human Genome Project
  - Identify all the 100,000 genes in human DNA
  - Determine the sequences of the 3 billion chemical base pairs of DNA
- The Internet applications
  - Quickly access and retrieve large amount of information such as Google Search

What kind of problem are solved by algorithms? (2/2)

- Electronic commerce with public-key cryptography and digital signatures
- Manufacturing and other commercial enterprises need to allocate scare resources in the most beneficial way.



1.2 Algorithms as a technology

- Efficiency:
  - Different algorithms solve the same problem often differ noticeably in their efficiency
  - These differences can be much more significant than difference due to hardware and software
- For example, in Chapter 2 we will see that *insertion sort takes time roughly equal* ***to*** c_1n^2 *(c_1 is constant) to sort n items. But, merge sort takes time roughly equal to* $c_2 n\lg n$ *(c_2 is constant)*

1.2 Algorithms as a technology

- For example, assume a faster computer A (10^{10} instructions/sec) running insertion sort against a slower computer B (10^7 instructions/sec) running merge sort.
- Suppose that $c_1=2$, $c_2=50$ *and* $n= 10^7$.
  - the execution time of computer A is $2(10^7)^2/ 10^{10}$ instructions/sec = 20,000seconds (more than 5.5 hours)
  - the execution time of computer B is $50 \cdot 10^7 \lg 10^7 / 10^7$ instructions/sec = 1,163seconds (less than 20 minutes)
- By using algorithm whose running time grows more slowly, Computer B runs 17 times faster than Computer A
- For *100 million* numbers
  - Insertion sort takes ≈ 23 days
  - Merge sort takes ≈ 4 hours