Algorithms

Definition (Informal)

An algorithm is a well-defined computational procedure that takes a set of values as input and produces a set of values as output.
Examples

- Human Genome Project:  
  db and data analysis

- Internet: data routing and search engines

- E-commerce:  
  crypto and digital signatures

- manufacturing:  
  scarce resource allocation

Algorithms as Technology

- system performance depends on choice of algorithm as much as on choice of hardware

- algorithms are at the core of most applications (GUI, OS, compiler, hardware)
Techniques for Algorithm Design

- incremental
- divide and conquer
- randomized algorithms
- dynamic programming
- greedy strategies
- backtracking
- branch-and-bound
Tractable (Efficient) Algorithms

- searching
- sorting
- order statistics
- graphs

Intractable (Inefficient) Algorithms

- Traveling Salesman Problem
- Satisfiability
- Hamiltonian Cycles
- Graph coloring
Algorithm Correctness

Is the algorithm "correct"? (Does it perform as advertised?)

Tools: preconditions and postconditions

Example: Linear Search

If at the start of execution, the following preconditions are true: list is an array of numElems $\geq$ 0 items indexed from 0 to numElems – 1 and val is the item sought,

then the algorithm terminates and the following postconditions are true: if val is in list, then position is equal to the index of the first occurrence of val in list, and otherwise position is equal to -1.

Does the algorithm work if the array is empty?
Algorithm Efficiency

- Algorithms use resources. Determining the space (memory, disk space) and the execution time of an algorithm is called algorithm analysis.

- Questions to Ask:
  What is the problem size? How can it vary?
  What is the Worst Case?
  What is the Average Case?
  How Much Space?
  How Much Time?
  How Much Time in the Average Case?
  Can We Do Better?
• Consider use of space and time as a function of the problem size.
  
  – If you triple the problem size, what happens to the use of space and time?
  
  – Assume each "basic" operation takes some constant amount of time. Tool: RAM Model
  
  – Count the number of times each operation is executed in the worst case.
  
  – Sum the times for each operation to get total time
Asymptotic Notation–the Language of Algorithm Analysis

Upper Bounds (Big-Oh)

• \( f \) is big-oh of \( g \)
• \( f = O(g) \)
• (a multiple of) \( g \) is (eventually) an upper bound for \( f \)

Lower Bounds (Big-Omega)

• \( f \) is big-omega of \( g \)
• \( f = \Omega(g) \)
• (a multiple of) \( g \) is (eventually) a lower bound for \( f \)
Asymptotically Tight Bounds (Big-Theta)

- \( f \) is big-theta of \( g \)
- \( f = \Theta(g) \)
- \( g \) is an asymptotically tight bound for \( f \)
- (some multiple of) \( g \) is (eventually) an upper bound for \( f \) and
- (another multiple of) \( g \) is (eventually) a lower bound for \( f \)