Algorithms

An algorithm is a finite set of precise instructions for performing a computation or for solving a problem.
Algorithm 1: Finding the Maximum Element in a finite Sequence

Procedure max(a_1, a_2, ..., a_n: integers)
Max := a_1
For i := 2 to n
    if max < a_i then max := a_i
{max is the largest element}
Algorithm 2

The Linear Search Algorithm

Procedure linear search(x: integer, a₁, a₂, ..., aₙ: distinct integers)

i := 1

While (i <= n and x ≠ aᵢ)
    i := i + 1

If i <= n then location := i
Else location := 0

{location is the subscript of the term that equals x, or is 0 if x is not found}
The Binary Search Algorithm

Procedure binary search (x: integer, a₁, a₂, ..., aₙ: increasing integers)
i:=1 {i is left endpoint of search interval}
j:= n {j is right endpoint of search interval}
While i < j
Beging
  m:= ⌊ (i+j)/2 ⌋
  if x > aₘ then i:=m+1
  else j:=m
End
If x = aᵢ then location := i
Else location := 0
{location is the subscript of the term equal to x, or 0 if x is not found}
Algorithm 4

The Bubble Sort

Procedure bubblesort($a_1, \ldots, a_n$: real numbers with $n \geq 2$)
For $i := 1$ to $n - 1$
    For $j := 1$ to $n - 1$
        if $a_j > a_{j+1}$ then interchange $a_j$ and $a_{j+1}$

\{ $a_1, \ldots, a_n$ is in increasing order \}

\textbf{The Bubble sort is the one of the simplest sorting algorithms but not one of the most efficient.}
Algorithm 6
Greedy Change-Making Algorithm

Procedure change($c_1, ..., c_n$): values of denominations of coins, where $c_1 > c_2 > ... > c_r$; n: a positive integer)

For $i := 1$ to $r$

    while $n \geq c_i$

    begin

        add a coin with value $c_i$ to the change

        $n := n - c_i$

    end
Theorem

The greedy algorithm produces change using the fewest coins possible.