Algorithms.

An algorithm is a finite sequence of precise instructions. Usually, it is used for performing a computation or solving a problem.

Example:

Algorithm to find the maximum element in a finite seq:

PROCEDURE max ($a_1, \dots, a_n$: real numbers)

$max \leftarrow a_1$

For $i \in [2, n]$:

    if $max < a_i$, then $max \leftarrow a_i$

RETURN max.

What it does: first, set the temporary maximum equal to the first element in a sequence. Then, compares the next element to the temporary maximum. If it is larger, we update max.

Algorithms will be written in pseudocode form (English language + computer language)

General properties of algorithms :

Input, output, definiteness, correctness, finiteness, effectiveness, generality.

Searching algorithms :

They deal with finding an element in an ordered list.

Linear search algorithm

PROCEDURE linear search ($x \in \mathbb{R}, (a_1, \dots, a_n) \in \mathbb{R}^n$)

$i \leftarrow 1$

WHILE $i \leq n$ and $x \neq a_i$:

$i \leftarrow i+1$

If $i \leq n$, then location $\leftarrow i$

Else location $\leftarrow 0$

RETURN location.

When the list is ordered, we have a more efficient algorithm :

Binary search algorithm

PROCEDURE binary search $(x \in \mathbb{R}, (a_1, \ldots, a_n) \in \mathbb{R}$ in increasing order$)$

$i \leftarrow 1$ (i is the left endpoint of search interval)

$j \leftarrow n$ (j is the right endpoint of search interval)

WHILE $i < j$:

$m \leftarrow \left\lfloor \frac{i+j}{2} \right\rfloor$

If $x > a_m$, then $i \leftarrow m+1$

Else $j \leftarrow m$.

If $x = a_i$, then location $\leftarrow i$

Else location $\leftarrow 0$.

Return location.
