Sorting algorithms

They are used to put the elements of a list in a given order. For example, given a list of n integers, a sorting algorithm puts them in increasing order.

The bubble sort algorithm :

PROCEDURE bubblesort ($a_1, \cdots, a_n$ real numbers, $n \geq 2$)

FOR i in $[1, n-1]$:

    FOR j in $[1, n-1]$:

        If $a_j > a_{j+1}$, then interchange a_j and a_{j+1}

RETURN list obtained after these operations.

Observe that in the first step the largest element of the list is in the last position. In the 2nd step, the 2nd largest is in the 2nd last position, and so on.

This shows that the bubble sort algorithm returns a increasing sequence of numbers at

the end.

Insertion sort:

PROCEDURE insertion sort ($a_1, \dots, a_n$: real numbers, $n \geq 2$):

① Compares a_2 and a_1.
It inserts a_2 before a_1 if $a_2 < a_1$.
It inserts a_2 after a_1 if $a_2 > a_1$.

② In the i-th step, compares a_i to $a_1, \dots, a_{i-1}$.
It inserts a_i before a_1 if $a_i < a_1$
It inserts a_i after a_{i-1} if $a_{i-1} < a_i$
It inserts a_i in between a_j and a_{j+1}
if $a_j \leq a_i \leq a_{j+1}$.

Greedy algorithms: algorithms that make what seems to be the best choice at each step.

Example:

Consider the problem of making n cents change with quarters (25 cents), dimes (10 cents), nickels (5 cents) and pennies (1 cents)
using the least total number of coins

using the least total ... of coins.

Strategy: choose the coin with the largest value at each step without exceeding n cents.

Example:

67 cents = 25·2 + 10 + 5 + 2 (total of 6 coins)

1st: select 25 (42 left)
2nd: select 25 (17 left)
3rd: select 10 (7 left)
4th: select 5 (2 left)
5th: select 2 coins of 1 cent

Greedy Change-making algorithm:

PROCEDURE change($c_1, \dots, c_r$: values of coins; $n \in \mathbb{N}$)
$c_1 < \dots < c_r$

For $i \in [1, r]$:

$d_i \leftarrow 0$ (it counts how many coins with value c_i are used.)

while $n \geq c_i$:

$d_i \leftarrow d_i + 1$

$n \leftarrow n - c_i$

Output: $(d_1, \dots, d_r)$.

we can show that the greedy algorithm gives the optimal solution for this problem.
