Lab 1 Cormen Exercises- Algorithms

Alberto Nicolai Romero Martinez

September 25, 2018

1 Exercise 2.1-1

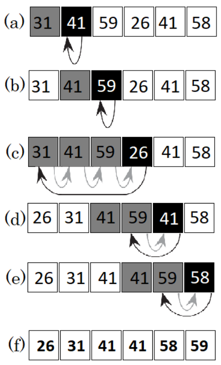

Figure 1: Insertion-Sort procedure with the array $A = 31, 41, 59, 26, 41, 58$ as the input.

2 Exercise 2.1-2

Insertion-Sort Backwards(A)

```
for $j = 2$ to $A.length$ do
    $key = A[j]$
    $i = j - 1$
    while $i > 0$ and $A[i] < key$ do
        $A[i + 1] = A[i]$
        $i = i - 1$
    end while
    $A[i + 1] = key$
    repeat
```

```
end for
```

3 Exercise 2.1-3

Searching Problem

```
position = NIL
for j = 0 to A.length − 1 do
  if A[k] = v then
    position = NIL
  else
    repeat
    end for
    return position
```

4 Exercise 2.1-4

Asumimos que cada arreglo de bits está ordenado de tal manera en que a mayor posición el bit es más significativo. Siendo A[0] y B[0] los bits menos significativos de cada entero, mientras A[n-1] y B[n-1] son los más significativos.
Entonces tenemos:
Binary sum

```
Integer[] C = new Integer[n + 1]
carry = 0
for j = 0 to A.length do
  x := A[j] + B[j] + carry
  C[j] = x mod 2
  if x>2 then
    carry = 1
  else
    carry = 0
  end if
end for
C[A.length] = carry
return C
```