

Algorithm. Finding the Maximum Element in a Finite Sequence.

Input: n [positive integer], a_1, a_2, \dots, a_n [integers]

Algorithm Body:

```
max :=  $a_1$ 
for  $i := 2$  to  $n$ 
    if  $max < a_i$  then  $max := a_i$ 
next  $i$ 
```

Output: max [integer]

Algorithm. Linear Search Algorithm.

Input: n [positive integer], a_1, a_2, \dots, a_n [distinct integers], x [integer]

Algorithm Body:

```
 $i := 1$ 
while ( $i \leq n$  and  $x \neq a_i$ )
     $i := i + 1$ 
end while
if  $i \leq n$ 
    then  $location := i$ 
    else  $location := 0$ 
```

Output: $location$ [integer]

[$location$ is the subscript of term equals x , or is 0 if x is not found]

Algorithm. Binary Search Algorithm.

Input: n [positive integer], a_1, a_2, \dots, a_n [increasing integers], x [integer]

Algorithm Body:

```
 $location := 0, bot := 1, top := n$ 
while ( $top \geq bot$  and  $location = 0$ )
     $m := \text{floor}((bot + top)/2)$ 
    if  $a_m = x$  then  $location := m$ 
    if  $a_m > x$ 
        then  $top := m - 1$ 
        else  $bot := m + 1$ 
end while
```

Output: $location$ [integer]

[$location$ is the subscript of term equal to x , or 0 if x is not found]

Algorithm. Division Algorithm.
(Positive divisor, nonnegative dividend.)

Input: a [nonnegative integer], d [positive integer]

Algorithm Body:

```
 $r := a$   
 $q := 0$   
while ( $r \geq d$ )  
     $r := r - d$   
     $q := q + 1$   
end while
```

Output: q, r [nonnegative integers]
[q is the quotient and r the remainder when a is divided by d . $a = dq + r$, $0 \leq r < d$]

Algorithm. Computing gcds by Subtraction.
(Finds greatest common divisor of two positive integers.)

Input: A, B [positive integers]

Algorithm Body:

```
 $a := A$   
 $b := B$   
while ( $a \neq 0$  and  $b \neq 0$ )  
    if  $a \geq b$   
        then  $a := a - b$   
        else  $b := b - a$   
    end while  
if  $a = 0$   
    then  $gcd := b$   
    else  $gcd := a$ 
```

Output: gcd [a positive integer]