

# Análisis de algoritmos recursivos

Pedro O. Pérez M., PhD.

Análisis y diseño de algoritmos  
Tecnológico de Monterrey

*pperezm@tec.mx*

01-2020

# Contenido

Algoritmos recursivos

Calculando el  $O(n)$  de algoritmos conocidos

# Algoritmos recursivos

- ▶ Este tipo de algoritmos no se pueden analizar como se hace con los algoritmos iterativos. El análisis de este tipo de algoritmos se hace utilizando el proceso inductivo.
- ▶ ¿Qué recuerdas del proceso inductivo? ¿Qué pasos hay que considerar?  
¿Cómo se realiza?

# Algoritmos conocidos

---

**Procedure 1** POW - Calculate  $x^n$  using a recursive algorithm.

---

**Input:**  $x : Real, n : Integer$

if  $n = 0$  then

return 1

else

return  $x * POW(x, n - 1)$

end if

---

---

## Procedure 2 ENIGMA - Unknown

---

Input:  $n : \text{Integer}$

if  $n \leq 0$  then

return 1

else

return  $ENIGMA(n - 1) + ENIGMA(n - 1)$

end if

---

---

**Procedure 3** BINARY\_SEARCH - Find the position of an element within an array using a recursive algorithm.

---

**Input:**  $A : \text{Array}$ ,  $low : \text{Index}$ ,  $high : \text{Index}$ ,  $key : \text{Object}$

    if  $(high - low + 1) = 1$  then

        return  $low$

    else

        NEXTSLIDE

    end if

---

---

```
mid  $\leftarrow$  (high + low)/2
if key = A[mid] then
    return mid
else if key < A[mid] then
    BINARY_SEARCH(A, low, mid - 1, key)
else
    BINARY_SEARCH(A, mid + 1, high, key)
end if
```

---

---

**Procedure 4** SEARCH\_BST - Find the position of an element within a binary search tree.

---

**Input:** *node* : *BSTNode*, *key* : *Object*

```
if key = node  $\rightarrow$  value then
    return true
else if value < node  $\rightarrow$  value then
    return SEARCH_BST(node  $\rightarrow$  left)
else if value > node  $\rightarrow$  value then
    return SEARCH_BST(node  $\rightarrow$  right)
end if
```

---



## Ejercicios de revisión I

---

**Procedure 5** POW2 -  $y = x^n$ , using a recursive algorithm

---

**Input:**  $x$  : *Real*,  $n$  : *Integer*

```
if  $n < 0$  then
    return POW2( $1/x$ ,  $-n$ )
else if  $n == 0$  then
    return 1
else if  $n == 1$  then
    return  $x$ 
else if  $n \bmod 2 = 0$  then
    return POW2( $x * x$ ,  $n/2$ )
else if  $n \bmod 2 = 1$  then
    return  $x * \text{POW2}(x * x, (n - 1)/2)$ 
end if
```

---

**Procedure 6 SUM\_LIST** - Calculate the sum of all the elements in a list of integers.

---

**Input:**  $n : \text{node\_list}$

if  $n \neq \text{NIL}$  then

return 0

else

return  $n \rightarrow \text{value} + \text{SUM\_LIST}(n \rightarrow \text{next})$

end if

---

# Merge Sort

---

## Procedure 7 COPY

---

**Input:**  $A, B$  : Array;  $low, high$  : Index  
  **for**  $i \leftarrow low$  **to**  $high$  **do**  
     $A[i] \leftarrow B[i]$   
  **end for**

---

---

## Procedure 8 MERGE

---

**Input:**  $A, B$  : Array;  $low, mid, high$  : Index

$i \leftarrow low$

$j \leftarrow mid + 1$

$k \leftarrow low$

**while**  $i \leq mid$  **and**  $j \leq high$  **do**

**if**  $A[i] < A[j]$  **then**

$B[k] = A[i]$

$i \leftarrow i + 1$

**else if**  $A[i] \geq A[j]$  **then**

$B[k] = A[j]$

$j \leftarrow j + 1$

**end if**

$k \leftarrow k + 1$

**end while**

---

---

```
while  $j \leq high$  do
   $B[k] = A[j]$ 
   $k \leftarrow k + 1$ 
   $j \leftarrow j + 1$ 
end while
while  $i \leq mid$  do
   $B[k] = A[i]$ 
   $k \leftarrow k + 1$ 
   $i \leftarrow i + 1$ 
end while
```

---

---

## Procedure 9 SPLIT

---

**Input:**  $A, B$  : Array;  $low, high$  : Index

**if**  $(high - low + 1) = 1$  **then**

**return**

**else**

$mid \leftarrow (high + low)/2$

$SPLIT(A, B, low, mid)$

$SPLIT(A, B, mid + 1, high)$

$MERGE(A, B, low, mid, high)$

$COPY(A, B, low, high)$

**end if**

---