

UT2 - PD2

Ejercicio 1.

Método Factorial (numero)

comienzo

// caso base: factorial de 0 es 1

Si numero = 0

$O(1)$

Devolver 1

$O(1)$

Fin si

Devolver numero * Factorial(numero - 1) $O(N)$

Fin

Caso base: factorial de 0 es 1

Si numero = 0 entonces

Devolver 1

Fin si

Verificación que el algoritmo procese el caso base:
averiguar factorial de 4:

1) Factorial(4) $\rightarrow 4 \times \text{Factorial}(3)$

2) Factorial(3) $\rightarrow 3 \times \text{Factorial}(2)$

3) Factorial(2) $\rightarrow 2 \times \text{Factorial}(1)$

4) Factorial(1) $\rightarrow 1 \times \text{Factorial}(0)$

5) Factorial(0) \rightarrow Retorna 1 \rightarrow caso base

2. El orden es $O(N)$

3. Si el número es negativo entra en una recursión infinita ya que cada vez los números se vuelven más negativos, lo cual produce un desbordamiento de la pila.

Ejercicio 2.

SumaLineal(A, n)

Si $n = 1$ entonces $O(1)$

Devolver $A[0]$ $O(1)$

Fin si

Devolver $\text{SumaLineal}(A, n-1) + A[n-1]$ $O(N)$

Fin

Caso base: n es igual a 1

Si $n = 1$ entonces

Devolver $A[0]$

Fin si

Verificación funcionamiento algoritmo

$n = 3$

1) $\text{SumaLineal}(A, 3) \rightarrow \text{SumaLineal}(A, 2) + A[2]$

2) $\text{SumaLineal}(A, 2) \rightarrow \text{SumaLineal}(A, 1) + A[1]$

3) $\text{SumaLineal}(A, 1) \rightarrow \text{Devolver } A[0]$

2. Orden $O(N)$

3. Si n es negativo el algoritmo entra en una recursión infinita ya que n se hace cada vez más negativo, lo cual desborda la pila.

Ejercicio 3.

Método Potencia (numero, exponente)

Si exponente = 0 entonces $O(1)$

Devolver 1 $O(1)$

Sino

Si exponente > 0 entonces $O(1)$

Devolver numero * Potencia(numero, exponente - 1) $O(N)$

Fin si

Sino

Devolver 1 / (numero * Potencia(numero, -exponente - 1) $O(N)$)

Caso base cualquier numero elevado a la 0 da resultado 1

Si exponente = 0 entonces

Devolver 1

Fin si

Verificar algoritmo llega a caso base

numero = 2 exponente = 3

1) Potencia(2, 3) $\rightarrow 2 * \text{Potencia}(2, 2)$

2) Potencia(2, 2) $\rightarrow 2 * \text{Potencia}(2, 1)$

3) Potencia(2, 1) $\rightarrow 2 * \text{Potencia}(2, 0)$

4) Potencia(2, 0) $\rightarrow \text{Devolver } 1$

2. Orden $O(N)$

Ejercicio 4

Metodo InvertirArray (A, i, j)

Si $i \geq j$ entonces $O(1)$

Retornar $O(1)$

Fin si

$aux \leftarrow vector[i]$ $O(1)$

$A[i] \leftarrow A[j]$ $O(1)$

$A[j] \leftarrow aux$ $O(1)$

invertirArray ($A, i+1, j-1$) $O(N)$

Fin

Caso base : Ya esta invertido.

Si $i \geq j$ entonces

Retornar

Verificar algoritmo procesa caso base

$i=0$ $j=4$

0, 1, 2, 3, 4

1) InvertirArray ($A, 1, 3$)

4, 3, 2, 1, 0

2) InvertirArray ($A, 2, 2$) \rightarrow Termina

2. Orden $O(N)$