ANÁLISIS DE ALGORITMOS

EJERCICIOS

Búsqueda Secuencial


```
función Secuencial(A[1..n]: vector de enteros; x, n: enteros) retorna (i: entero)
    var i : entero fvar
    i=1;
    mientras ( i ≤ n y A[ i ] ≠ x ) hacer
            i = i + 1;
    fmientras
    retorna i;
ffunción
```

$$T_{MC}(n) = 2 \in \theta(1)$$

$$T_{PC}(n) = 2 + \sum_{i=1}^{n} 1 = 2 + n \in \theta(n)$$

$$T(n) \in \Omega(1) \wedge T(n) \in O(n)$$

Visto en clase (talla **n** y existe mejor y peor caso). Falta hacer los cálculos y usar la notación.

EJEMPLO 1

¿ Hay Mejor y Peor caso ?

```
Procedimiento DosA (A[1..n]: vector; n: entero)
   var suma=0, producto=1: entero fvar
   si n ≥ 2 entonces
      para i=1 hasta n hacer
         suma= suma + i;
      fpara
   si no
      para i=1 hasta n hacer
         suma = suma + i;   producto=producto * i;
      fpara
   fsi
fprocedimiento
```

$\boxed{NO}$

```
Procedimiento DosB ( A[1..n]: vector; n: entero)
   var suma=0, producto=1: entero fvar
   si A[1] ≥ 2 entonces
      para i=1 hasta n DIV 2 hacer
         suma= suma + i;
      fpara
   si no
      para i=1 hasta n hacer
         suma= suma + i;   producto=producto * i;
      fpara
   fsi
fprocedimiento
```

$\boxed{SÍ}$

EJEMPLO 2

¿ Hay Mejor y Peor caso ?

…

si n es par entonces

para i=1 hasta n hacer

s=s+1;

finpara

$\in \theta(\boldsymbol{n})$

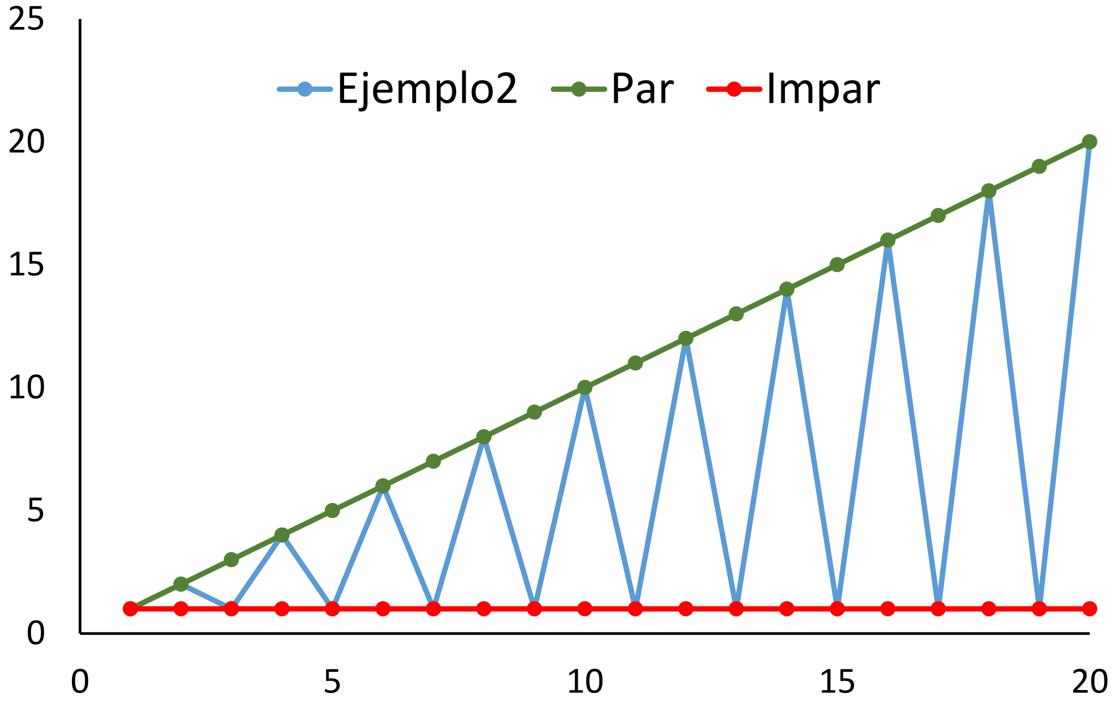
si no

s=s+1;

fsi

$\in \theta(\mathbf{1})$

…



$$\boxed{T(n) \in \Omega(1) \wedge T(n) \in O(n)}$$

EJEMPLO 3


```
función  Ejemplo3 (A[1..n]: vector de enteros; n: entero) retorna (s: entero)
    var s:entero fvar
    s = 0;
    para i = 1 hasta n hacer
            s = s + A[i];
    fpara
    retorna s;
ffunción
```

Talla del problema: **n**

Existe mejor y peor caso: **No**

$$T(n) = 2 + \sum_{i=1}^{n} 1 = 2 + n \in \theta(n)$$

EJEMPLO 4


```
procedimiento  Ejemplo4 (A[1..n][1..m], B[1..n][1..m]: matriz de enteros; n, m: entero)
    para i = 1 hasta n hacer
            para j = 1 hasta m hacer
                    A[i][j] = B[i][j] + A[i][j];
            fpara
    fpara
fprocedimiento
```

Talla del problema: **(n, m)**

Existe mejor y peor caso: **No**

$$T(n,m) = \sum_{i=1}^{n}\left(\sum_{j=1}^{m} 1\right) = \sum_{i=1}^{n} m = nm \in \theta(nm)$$

EJEMPLO 5


```
función  Ejemplo5 (n: entero) retorna (x: entero)
    var i, j, x : entero fvar
    i = n;    j = n;   x=0;
    mientras i ≠ 0 hacer
            i = i - 1;
            mientras j ≠ 0 hacer
                    j = j – 1;
                    x = x + i + j;
            fmientras
    fmientras
    retorna x;
ffunción
```

Talla del problema: **n**

Existe mejor y peor caso: **No**

$$T(n) = 4 + \sum_{i=1}^{n} 1 + \sum_{j=1}^{n} 2 = 4 + n + 2n \in \theta(n)$$

EJEMPLO 6


```
función  Ejemplo6 (n: entero) retorna (x: entero)
    var i, j, x : entero fvar
    i = n;   x=0;
    mientras i ≠ 0 hacer
            i = i - 1;    j = n;
            mientras j ≠ 0 hacer
                    j = j – 1;
                    x = x + i + j;
            fmientras
    fmientras
    retorna x;
ffunción
```

Talla del problema: **n**

Existe mejor y peor caso: **No**

$$T(n) = 3 + \sum_{i=1}^{n}\left(2 + \sum_{j=1}^{n} 2\right) = 3 + \sum_{i=1}^{n} 2 + \sum_{i=1}^{n} 2n = 3 + 2n + 2n^2 \in \theta(n^2)$$

EJEMPLO 7


```
función  Ejemplo7(n: entero) retorna (x: entero)
    var j, x : entero fvar
    j = n;   x=0;
    mientras j ≠ 0 hacer
            x = x + 2j;
            j = j div 2;
    fmientras
    retorna x;
ffunción
```

n	Número de iteraciones	Valores de j
1	1	{1}
2	2	{2, 1}
3	2	{3, 1}
4	3	{4, 2, 1}
5	3	{5, 2, 1}
6	3	{6, 3, 1}
7	3	{7, 3, 1}
8	4	{8, 4, 2, 1}

Talla del problema: **n**

Existe mejor y peor caso: **No**

EJEMPLO 7

1ª iteración → $n \equiv n/2^0$

2ª iteración → $n \equiv n/2^1$

3ª iteración → $n \equiv n/2^2$

…

$$\boxed{T(n) = 3 + \sum_{i=0}^{\log_2 n} 2 = 3 + 2(\log_2 n + 1) = 5 + 2\log_2 n \in \theta(\log_2 n)}$$

i+1ª iteración → $n \equiv n/2^i$

$$\frac{n}{2^i} = 1 \rightarrow n = 2^i \rightarrow \log_2 n = i \rightarrow Numero\ de\ Iteraciones = 1 + i = 1 + \log_2 n$$

EJEMPLO 8


```
procedimiento  Ejemplo8 (A[1..n], B[1..n]: vectores de enteros; n: entero)
    para i = 1 hasta n hacer
            B[i] = Suma(A, n);
    fpara
fprocedimiento
función  Suma (A[1..n]: vector de enteros; n: entero) retorna (s: entero)
    var s : entero fvar
    s = 0;
    para i = 1 hasta n hacer
            s = s + A[i];
    fpara
    retorna s;
ffunción
```

Talla del problema: **n**

Existe mejor y peor caso: **No**

$$T(n) = \sum_{i=1}^{n}(2+n) = \sum_{i=1}^{n} 2 + \sum_{i=1}^{n} n = 2n + n^2 \in \theta(n^2)$$

EJEMPLO 9


```
procedimiento  Ejemplo9 (A[1..n], B[1..n]: vectores de enteros; n: entero)
    para i = 1 hasta n hacer
            B[i] = Suma(A, i);
    fpara
fprocedimiento
función  Suma (A[1..n]: vector de enteros; n: entero) retorna (s: entero)
    var s : entero fvar
    s = 0;
    para i = 1 hasta n hacer
            s = s + A[i];
    fpara
    retorna s;
ffunción
```

Talla del problema: **n**

Existe mejor y peor caso: **No**

$$T(n) = \sum_{i=1}^{n}(2+i) = \sum_{i=1}^{n} 2 + \sum_{i=1}^{n} i = 2n + \frac{n(n+1)}{2} = \frac{5n+n^2}{2} \in \theta(n^2)$$

EJEMPLO 10


```
función Ejemplo10 (A[1..n]: vector de enteros; n: entero) retorna (p: entero)
    var i, pos : entero fvar
    p = 1;
    para i = 2 hasta n hacer
            si A[i] < A[p] entonces p = i; fsi
    fpara
    retorna p;
ffunción
```

Talla del problema: **n**

Existe mejor y peor caso: **Sí**

$$T_{MC}(n) = 2 + \sum_{1}^{n} 1 = 2 + n \in \theta(n)$$

$$T_{PC}(n) = 2 + \sum_{i=1}^{n} 1 = 2 + n \in \theta(n)$$

$$T(n) \in \Omega(n) \wedge T(n) \in O(n) \Longrightarrow T(n) \in \theta(n)$$

EJEMPLO 11

$\{ n \geq 0 \}$

```
función Ejemplo11 (A[1..n]: vector de enteros; n, x: entero) retorna (entero)
    si n = 0 entonces
        retorna 0;
    si no
        si A[n] = x entonces
            retorna n;
        si no
            retorna Ejemplo11 (A, n-1, x);
        fsi
    fsi
ffunción
```

Talla del problema: **n**

Existe mejor y peor caso: **Sí**

$$T_{MC}(n) = \begin{cases} c_1 & si\ (n = 0) \\ c_2 & si\ (n > 0) \end{cases} \Longrightarrow T_{MC}(n) \in \theta(1)$$

$$T_{PC}(n) = \begin{cases} c_1 & si\ (n = 0) \\ T_{PC}(n-1) + c_2 & si\ (n > 0) \end{cases} \Longrightarrow T_{pC}(n) \in \theta(n)$$

EJEMPLO 12

$\{ n > 0 \}$

```
procedimiento Ejemplo12 (A, B, C: torres; n: entero)
    si n = 1 entonces
            Mover(A, C);
    si no
            Ejemplo12(A, C, B, n -1);
            Mover(A, C);
            Ejemplo12(B, A, C, n - 1);
    fsi
fprocedimiento
```

Talla del problema: **n**

Existe mejor y peor caso: **No**

$$T(n) = \begin{cases} c_1 & si\ (n = 1) \\ 2T(n-1) + c_2 & si\ (n > 1) \end{cases} \Longrightarrow T(n) \in \theta(2^n)$$

donde *Mover(A, C)* tiene coste $\theta(1)$

EJEMPLO 13


```
procedimiento Ejemplo13 (A[1..n]: vector de enteros; i, j: entero)
    si i < j  entonces
            Ejemplo13(A, i, (i + j) div2);
            Ejemplo13(A, (i + j) div2 + 1, j);
            Mezcla(A, i, j);
    fsi
fprocedimiento
```

Talla del problema: **j - i + 1**

Existe mejor y peor caso: **No**

$$T(n) = \begin{cases} c_1 & si\ (n \leq 1)\ \ (\equiv i \geq j) \\ 2T(n\ div\ 2) + nc_2 + c_3 & si\ (n > 1)\ \ (\equiv i < j) \end{cases} \Longrightarrow T(n) \in \theta(n \log_2(n))$$

donde *Mezcla*$(A, i, j) \in \theta(j - i + 1)$ y la llamada inicial a la función es: *Ejemplo13(A, 1, n)*

EJEMPLO 14


```
función Ejemplo14 (A[1..n]: vector de enteros; i, j: entero) retorna(booleano)
    si  i ≥ j entonces
            retorna Cierto;
    si no
            si A[i] = A[j] entonces
                    retorna Ejemplo14(A, i + 1, j - 1);
            si no
                    retorna Falso;
            fsi
    fsi
ffunción
```

Llamada inicial: $Ejemplo14(A, 1, n)$

Talla del problema: **n**

Existe mejor y peor caso: **Sí**

$$T_{MC}(n) = \begin{cases} c_1 & si\ (n \leq 1) \\ c_2 & si\ (n > 1) \end{cases} \Longrightarrow T_{MC}(n) \in \theta(1)$$

$$T_{PC}(n) = \begin{cases} c_1 & si\ (n \leq 1) \\ T_{PC}(n-2) + c_2 & si\ (n > 1) \end{cases} \Longrightarrow T_{pC}(n) \in \theta(n)$$

EJEMPLO 15

$\{ n > 0 \}$

```
función Ejemplo15 (n: entero) retorna (entero)
    si n=1 ∨ n=2 entonces
            retorna 1;
    si no retorna
            Ejemplo15(n-1) + Ejemplo15(n-2);
    fsi
ffunción
```

Talla del problema: **n**

Existe mejor y peor caso: **No**

$$T_1(n) = \begin{cases} c_1 & si\ (n \leq 2) \\ 2T_1(n-2) + c_2 & si\ (n > 2) \end{cases} \quad y \quad T_2(n) = \begin{cases} c_1 & si\ (n \leq 2) \\ 2T_2(n-1) + c_2 & si\ (n > 2) \end{cases}$$

$$T(n) \in \Omega\left(2^{n/2}\right) \qquad T(n) \in O(2^n)$$

Ejemplo 16


```
función Ejemplo16 (A[1..n]: vector de enteros; x, i, j: entero) retorna (entero)
    var m : entero fvar
    si i > j entonces  retorna i;
    si no
       m = (i+j) div 2;
       si x = A[m] entonces retorna m;
       si no
          si x > A[m] entonces
             retorna Ejemplo16(A, x, m+1, j);
          si no
             retorna Ejemplo16(A, x, i, m-1);
          fsi
       fsi
    fsi
ffunción
```

Talla del problema: **j - i + 1**

Existe mejor y peor caso: **Sí**

$$T_{MC}(n) = \begin{cases} c_1 & si\ (n = 0) \\ c_2 & si\ (n > 0) \end{cases} \Longrightarrow T_{MC}(n) \in \theta(1)$$

$$T_{PC}(n) = \begin{cases} c_1 & si\ (n = 0) \\ T_{PC}(n\ div\ 2) + c_2 & si\ (n > 0) \end{cases} \Longrightarrow T_{pC}(n) \in \theta(\log n)$$

A está ordenado en sentido ascendente. Llamada inicial: *Ejemplo16(A, x, 1, n)*

EJEMPLO 17

$\{ n \geq 0 \}$

```
función  Ejemplo17 (A[1..n]: vector de enteros; n: entero) retorna (entero)
    si n≤1 entonces
        retorna 1;
    si no
        retorna  Ejemplo17(A, n-1) + Ejemplo17(A, n-2) + Ejemplo17(A, n DIV 2);
    fsi
ffunción
```

Talla del problema: **n**

Existe mejor y peor caso: **No**

$$T(n) = \begin{cases} c_1 & si\ (n \leq 1) \\ T(n-1) + T(n-2) + T(n\ div\ 2) + c_2 & si\ (n > 1) \end{cases}$$

$$T(n) \in \Omega(n^{1.58}) \wedge T(n) \in O(3^n)$$

EJEMPLO 18


```
función Ejemplo18 (A: lista de enteros) retorna (s: entero)
  var m, n, s : entero fvar
  m = 1; n = |A|; s = 0;
  mientras m ≤ n hacer
    si A_m ≠ 0 entonces
      para j = m hasta n hacer
        s = s + A_j;
      fpara
    fsi
    m = 2 * m;
  fmientras
  retorna s;
ffunción
```

Talla del problema: **n** ($|A|$)

Existe mejor y peor caso: **SÍ**

$$T_{MC}(n) = 3 + \sum_{i=0}^{\log n} 1 \qquad \in \theta(\log n)$$

$$T_{PC}(n) = 3 + \sum_{i=0}^{\log n} \left(1 + \sum_{j=m}^{n} 1 \right) \in \theta(n \log n)$$

Las operaciones $|A|$ ($\equiv$ número de elementos de la lista), $A_m \neq 0$ y $s = s + A_j$ tienen un coste constante, esto es pertenecen a $\theta(1)$

EJEMPLO 19


```
función Ejemplo19 (A[1..n]: vector de enteros; x, n: entero) retorna (entero)
   var i, m, j:entero; encontrado : booleano fvar
   i=1; j=n; encontrado = Falso
   mientras ( i ≤ j ∧ encontrado = Falso ) hacer
      m = (i+j) div 2;
      si A[m] = x entonces encontrado = Cierto;
      si no  si A[m] > x entonces j = m-1;
             si no i = m+1;
             fsi
      fsi
   fmientras
   si (encontrado = cierto) entonces retorna m; si no retorna i; fsi
ffunción
```

donde el vector A está ordenado en sentido ascendente.

Talla del problema: **n**

Existe mejor y peor caso: **SÍ**

$T_{MC}(n) \in \theta(1)$

$T_{PC}(n) \in \theta(\log n)$