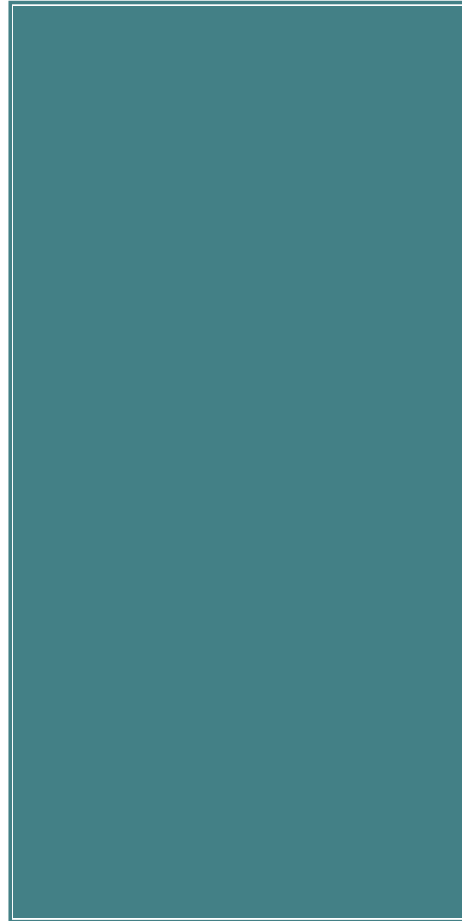


# ANÁLISIS DE ALGORITMOS

1

## EJERCICIOS



# BÚSQUEDA SECUENCIAL

2

función Secuencial( $A[1..n]$ :vector de enteros;  $x, n$ : enteros) retorna ( $i$ : entero)

var  $i$  : entero fvar

$i=1$ ;

mientras (  $i \leq n$  y  $A[i] \neq x$  ) hacer

$i = i + 1$ ;

fmientras

retorna  $i$ ;

ffunción

Tu texto aquí 1

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

# EJEMPLO 1

3

## ¿ Hay Mejor y Peor caso ?

Procedimiento DosA (A[1..n]: vector; n: entero)

var suma=0, producto=1: entero fvar

si  $n \geq 2$  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

Procedimiento DosB ( A[1..n]: vector; n: entero)

var suma=0, producto=1: entero fvar

si  $A[1] \geq 2$  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

# EJEMPLO 2

4

¿ Hay Mejor y Peor caso ?

...

si n es par entonces

para i=1 hasta n hacer

s=s+1;

finpara

si no

s=s+1;

fsi

...

# EJEMPLO 3

5

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

# EJEMPLO 4

6

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

# EJEMPLO 5

7

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

# EJEMPLO 6

8

función Ejemplo6 (n: entero) retorna (x: entero)

var i, j, x : entero fvar

i = n; x=0;

mientras i  $\neq$  0 hacer

i = i - 1; j = n;

mientras j  $\neq$  0 hacer

j = j - 1;

x = x + i + j;

fmientras

fmientras

retorna x;

ffunción



# EJEMPLO 7

9

función Ejemplo7 (n: entero) retorna (x: entero)

var j, x : entero fvar

j = n; x=0;

mientras j  $\neq$  0 hacer

x = x + 2j;

j = j div 2;

fmientras

retorna x;

ffunción

# EJEMPLO 8

10

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

# EJEMPLO 9

11

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

# EJEMPLO 10

12

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

# EJEMPLO 11

13

$\{ n \geq 0 \}$

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

# EJEMPLO 12

14

$\{ n > 0 \}$

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

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

# EJEMPLO 13

15

procedimiento Ejemplo13 ( $A[1..n]$ : vector de enteros;  $i, j$ : entero)

  si  $i < j$  entonces

    Ejemplo13( $A, i, (i + j) \text{div} 2$ );

    Ejemplo13( $A, (i + j) \text{div} 2 + 1, j$ );

    Mezcla( $A, i, j$ );

  fsi

fprocedimiento

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

# EJEMPLO 14

16

función Ejemplo14 ( $A[1..n]$ : vector de enteros;  $i, j$ : entero) retorna(booleano)

si  $i \geq j$  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:  $Ejemplo14(A, 1, n)$



# EJEMPLO 15

17

$\{ n > 0 \}$

función Ejemplo15 (n: entero) retorna (entero)

si  $n=1 \vee n=2$  entonces

retorna 1;

si no retorna

Ejemplo15(n-1) + Ejemplo15(n-2);

fsi

ffunción

# EJEMPLO 16

18

```
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
```

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

# EJEMPLO 17

19

$\{ n \geq 0 \}$

función Ejemplo17 (A[1..n]: vector de enteros; n: entero) retorna (entero)

    si  $n \leq 1$  entonces

        retorna 1;

    si no

        retorna Ejemplo17(A, n-1) + Ejemplo17(A, n-2) + Ejemplo17(A, n DIV 2);

    fsi

ffunción

# EJEMPLO 18

20

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  $A_m \neq 0$  entonces

para j = m hasta n hacer

s = s +  $A_j$ ;

fpara

fsi

m = 2 \* m;

fmientras

retorna s;

ffunción

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

21

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 (  $i \leq j \wedge$  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.