Guía de Trabajos Prácticos

Unidad I
Complejidad Algorítmica

1. Calcule la complejidad de las siguientes funciones y procedimientos

    1.
    ```
    PROCEDIMIENTO intercambia (REF x : ENTERO, REF y : ENTERO) ES
        aux : ENTERO;
        aux := x;
        x := y;
        y := aux;
    FINPROCEDIMIENTO
    ```

    2.
    ```
    FUNCION max(x : ENTERO, y : ENTERO) : ENTERO ES
        result : ENTERO;
        SI (x >= y) ENTONCES
            result := x;
        SINO
            result := y;
        FINSI función
        DEVOLVER result;
    FINFUNCION
    ```

    3.
    ```
    FUNCION suma (v : VECTOR(ENTERO), tamaño : ENTERO): ENTERO ES
        i, result : ENTERO; result := 0;
        PARA i DESDE 0 HASTA tamaño-1 HACER
            result := result + v[i];
        FINPARA
        DEVOLVER result;
    FINFUNCION
    ```

    4.
    ```
    FUNCION aSaber(v : VECTOR(ENTERO),n: ENTERO): ENTERO ES
        i, result :ENTERO;
        result :=0;        i := 0;
        MIENTRAS (i< tamaño) HACER
            result:= result + v[i];
    ```

```
            i := i+1;
        FINMIENTRAS

        DEVOLVER result;

    FINFUNCION
```

5.

```
    FUNCION aSaber(v : VECTOR(ENTERO),n: ENTERO): ENTERO ES

        i, result : ENTERO;

        result := 0;   i := tamaño - 1;

        MIENTRAS (i >= 0) HACER
            result := result + v[i];

            i := i - 1;

        FINMIENTRAS

        DEVOLVER result;


    FINFUNCION
```

6.

```
FUNCION aSaber(v : VECTOR(ENTERO), n: ENTERO): ENTERO ES

    i, result : ENTERO;

    result := 0; i :=0;

    MIENTRAS (i < tamaño-1) HACER

        SI (v[i] <= v[i+1]) ENTONCES

            intercambia(v[i],v[i+1]);

        FINSI

    FINMIENTRAS

    DEVOLVER result;

FINFUNCION
```

NOTA: intercambia es la función del ejercicio del ítem (a).

7.

```
PROCEDIMIENTO inserción(v : VECTOR(ENTERO), tamaño : ENTERO) ES

    i, j : ENTERO;

    PARA i DESDE 0 HASTA tamaño - 2 HACER

        min : ENTERO; min := i;

        PARA j DESDE i+1 HASTA tamaño – 1 HACER

            SI (v[j] < v[min]) ENTONCES

                min := j;

            FINSI

        FINPARA

        intercambia(v[i],v[min])

    FINPARA
```

FINPROCEDIMIENTO

2. Dados los siguientes algoritmos, calcular su tiempo de ejecución (T(n)) en el Peor Caso, y decir qué complejidad o crecimiento asintótico tiene (O).

Siendo:

```
Constante n
Vector: arreglo [1 .. n] de enteros
```

a)

```
Procedimiento Algoritmo1 (vector a)
Ambiente
   Entero j, i, temp;
Inicio
   Para i:= 1 a n-1 hacer
         Para j:= n a i+1, -1 hacer
               Si a[j-1]> a[j] entonces
                     temp:= a[j-1];
                     a[j-1]:= a[j];
                     a[j]:= temp;
               fin_si;
         Fin_para;
   Fin_para;
Fin
```

b)

```
Función Algoritmo2 (vector a, entero c):
entero Ambiente
   Entero inf, sup, i;
Inicio
   inf:=1; sup:=n;
   Mientras (sup>=inf) hacer
         i:= (inf + sup) / 2;
         Si a[i]=c entonces Retorna i;
               sino
                     Si c < a[i] entonces sup:= i - 1;
                           sino
                                 inf:= i + 1;
                     fin_si;
         fin_si;
   Fin_mientras;
   Retorna 0;
Fin
```

c)

```
Función Algoritmo3 (entero n, m):
entero; Ambiente
   Entero temp;
Inicio
   Mientras m > 0 hacer
         temp:= m;
         m:= n MOD m;
         n:= temp;
   Fin_mientras;
   Retorna n;
Fin
```

d)

```
Procedimiento Algoritmo4 (entero n)
Ambiente
   Entero i, j, k, s;
Inicio
   S:= 0;
```

```
   Para i:= 1 a n-1 hacer
        Para j:= i+1 a n hacer
             Para k:= 1 a j hacer
                  S:= s+2;
             Fin_para;
        Fin_para;
   Fin_para;
Fin
```

3. Calcule, para cada uno de los ejercicios del ítem 2, cual es espacio en memoria reservado para las variables y estructuras utilizadas. Tome como referencia la tabla siguiente:

TIPOS PRIMITIVOS (sin métodos; no son objetos; no necesitan una invocación para ser creados)

NOMBRE	TIPO	OCUPA	RANGO APROXIMADO
byte	Entero	1 byte	-128 a 127
short	Entero	2 bytes	-32768 a 32767
int	Entero	4 bytes	$2*10^9$
long	Entero	8 bytes	Muy grande
float	Decimal simple	4 bytes	Muy grande
double	Decimal doble	8 bytes	Muy grande
char	Carácter simple	2 bytes	---
boolean	Valor true o false	1 byte	---