EJERCICIOS DE COMPLEJIDAD ALGORÍTMICA

1. Calcula la complejidad del siguiente procedimiento:

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

2. Calcular la complejidad de la siguiente función:

```
FUNCION max(x : ENTERO, y : ENTERO) : ENTERO ES

   result : ENTERO;
   SI (x >= y) ENTONCES
       result := x;
   SINO
       result := y;
   FINSI
   DEVOLVER result;
FINFUNCION
```

3. Calcular la complejidad de la siguiente función:

```
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. Calcular la complejidad de la siguiente función

```
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. Calcular la complejidad de la siguiente función

```
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. Calcular la complejidad de la siguiente función

```
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 - 2;
   FINMIENTRAS
   DEVOLVER result;
FINFUNCION
```

7. Calcular la complejidad de la siguiente función

```
FUNCION aSaber(v : VECTOR(ENTERO), n: ENTERO)  : ENTERO ES
   i, result : ENTERO;
   result := 0; i :=0;
   MIENTRAS ((i < tamaño) && ((v[i] DIV 2) != 0) HACER
      result := result + v[i];
      i := i + 1;
   FINMIENTRAS
   DEVOLVER result;
FINFUNCION
```

8. Calcular la complejidad de la siguiente función

```
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 1

9. Calcular la complejidad del siguiente procedimiento

```
PROCEDIMIENTO burbuja(v : VECTOR(ENTERO), tamaño : NATURAL) ES
   i, j : ENTERO;
   PARA i DESDE 2 HASTA tamaño - 1 HACER
      PARA j DESDE 0 HASTA tamaño – i HACER
         SI (v[j] > v[j+1]) ENTONCES
            intercambia(v[j],v[j+1]);
         FINSI
      FINPARA
   FINPARA
FINPROCEDIMIENTO
```

10. Calcular la complejidad del siguiente procedimiento.

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

11. Calcular la complejidad del siguiente procedimiento.

```
PROCEDIMIENTO inserción(v : VECTOR(ENTERO), tamaño : NATURAL) ES
   i, j : ENTERO;
   PARA i DESDE 0 HASTA tamaño - 2 HACER
      intercambia(v[i],v[posMinimo(v,i,tamaño-1)])
   FINPARA
FINPROCEDIMIENTO

FUNCION posMinimo(v : VECTOR(ENTERO), inicio, fin : NATURAL) ES
   i, result: ENTERO;
   result := inicio;
   PARA i DESDE inicio+1 HASTA fin HACER
      SI (v[i] < v[min]) ENTONCES
         result := i;
      FINSI
    FINPARA
    DEVOLVER result;
FINFUNCION
```

12. Calcular la complejidad del siguiente algoritmo recursivo tipo divide y vencerás

```
PROCEDIMIENTO mezcla (REF v : VECTOR(ENTERO), tamaño : NATURAL) ES
    mezclaRecursivo(v : VECTOR(ENTERO), 0,tamaño-1);
FINPROCEDIMIENTO

PROCEDIMIENTO mezclaRecursivo (REF v:VECTOR(ENTERO), inicio, fin : NATURAL)
    ES SI (inicio < fin) ENTONCES
       mezclaRecursivo(v,inicio, (inicio + fin) div 2);
       mezclaRecursivo(v,((inicio + fin) div 2)+1,fin);
       fusiona(v,inicio,(inicio + fin) div 2),fin)
    FINSI
FINFUNCION

PROCEDIMIENTO fusiona(REF v : VECTOR(ENTERO), inicio, medio, fin : NATURAL) ES
   aux : VECTOR DIM (fin – inicio + 1) DE ENTERO;
   i1,i2, pos : ENTERO; i1 := inicio; i2 := medio+1; pos := 0;
   MIENTRAS ((i1 <= medio) AND (i2 <= fin)) HACER
     SI (v[i1] <= v[i2]) ENTONCES
        aux[pos] := v[i1];
         i1 := i1 + 1;
     SINO
         aux[pos] := v[i2];
         i2 := i2 + 1;
     FINSI
     pos := pos + 1;
   FINMIENTRAS
```