



EJERCICIOS DE COMPLEJIDAD ALGORÍTMICA (CLASE)

1. Calcula la complejidad del siguiente algoritmo, asumiendo que todas las operaciones de la clase `Matriz` tienen un tiempo de ejecución $O(1)$.

```
FUNCION multiplica(x : Matriz(NATURAL),
                  y : Matriz(NATURAL)) : Matriz(NATURAL) ES

  i, j, k : NATURAL;
  result  : Matriz(NATURAL); result := NUEVO Matriz(x.getDimension());
  temp    : NATURAL;

  PARA i DESDE 0 HASTA x.getDimension()-1 HACER
    PARA j DESDE 0 HASTA x.getDimension()-1 HACER
      temp := 0;
      PARA k DESDE 0 HASTA x.getDimension()-1 HACER
        temp := temp + x.get(i,k)*y.get(k,j);
      FINPARA
      result.set(i,j) := temp;
    FINPARA
  FINPARA

  DEVOLVER result;
FINFUNCION
```

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

```
PROCEDIMIENTO invertir(REF v : VECTOR(Carácter), inicio, fin : NATURAL) ES

  SI (inicio < fin) ENTONCES
    SI (inicio == (fin -1)) ENTONCES
      aux : Carácter;
      aux := v[inicio];
      v[inicio] := v[fin];
      v[fin] := aux;
    SINO
      medio := (inicio+fin) div 2;
      invertir(v,inicio,medio);
      invertir(v,medio+1,fin);
    FINSI
  FINSI
  DEVOLVER result;
FINFUNCION
```

Pablo Sánchez Barreiro.