

1. Calcular la eficiencia del siguiente algoritmo:

```

1: int ejemplo1(int n)
2: {
3:     if (n <= 1)
4:         return 1;
5:     else
6:         return (ejemplo1(n - 1) + ejemplo1(n - 1));
7: }

```

2. Calcular la eficiencia del siguiente algoritmo:

```

1: int ejemplo2(int n)
2: {
3:     if (n == 1)
4:         return n;
5:     else
6:         return (ejemplo2(n/2) + 1);
7: }

```

3. Calcular la eficiencia del algoritmo de las Torres de Hanoi por expansión de la recurrencia.

```

Hanoi(origen,destino,pivote,discos):
    si discos=1
        moveruno(origen,destino)
    en otro caso
        Hanoi(origen,pivote,destino,discos-1)
        moveruno(origen,destino)
        Hanoi(pivote,destino,origen,discos-1)

```

4. Resolver las siguientes ecuaciones y dar su orden de complejidad:

$$a) \quad f(n) = \begin{cases} n & \text{si } n = 0 \text{ ó } n = 1 \\ f(n-1) + f(n-2) & \text{en otro caso} \end{cases}$$

$$b) \quad T(n) = \begin{cases} n & \text{si } n = 0, 1 \text{ ó } 2 \\ 5T(n-1) - 8T(n-2) + 4T(n-3) & \text{en otro caso} \end{cases}$$

$$c) \quad T(n) = \begin{cases} 0 & \text{si } n = 0 \\ 2T(n-1) + 1 & \text{en otro caso} \end{cases}$$

$$d) \quad T(n) = 4T\left(\frac{n}{2}\right) + n, \quad n > 1 \text{ y potencia de } 2$$

$$e) \quad T(n) = 4T\left(\frac{n}{2}\right) + n^2, \quad n > 1 \text{ y potencia de } 2$$

$$f) \quad T(n) = 2T\left(\frac{n}{2}\right) + n \cdot \log n, \quad n > 1 \text{ y potencia de } 2$$

$$g) \quad T(n) = \begin{cases} 1 & \text{si } n = 2 \\ 2T(\sqrt{n}) + \log \log n & \text{con } n \geq 4 \end{cases}$$

$$h) \quad T(n) = T\left(\frac{n}{2}\right) + T^2\left(\frac{n}{4}\right) \quad \text{sin caso base}$$