EFICIENCIA
RELACIÓN DE EJERCICIOS

1. Demostrar las siguientes propiedades:

   a. $\forall k>0$, $k \cdot f(n) \in O(f(n))$.

   b. $n^r \in O(n^k)$ si $0 \leq r \leq k$.

   c. $O(n^k) \subset O(n^{k+1})$

   d. $\forall b>1$, $k \geq 0$, $n^k \in O(b^n)$.

   e. $\forall b>1$, $k>0$, $\log_b n \in O(n^k)$.

   f. Si $f(n) \in O(g(n))$ y $h(n) \in O(g(n))$, entonces $f(n)+h(n) \in O(g(n))$.

   g. Si $f(n) \in O(g(n))$, entonces $f(n)+g(n) \in O(g(n))$.

   h. *Reflexividad:* $f(n) \in O(f(n))$.

   i. *Transitividad:* Si $f(n) \in O(g(n))$ y $g(n) \in O(h(n))$, entonces $f(n) \in O(h(n))$.

   j. *Regla de la suma:*
   Si T1(n) es O(f(n)) y T2(n) es O(g(n)),
   entonces T1(n) + T2(n) es O(max{f(n),g(n)}).

   k. *Regla del producto:*
   Si T1(n) es O(f(n)) y T2(n) es O(g(n)),
   entonces $T1(n) \cdot T2(n)$ es $O(f(n) \cdot g(n))$.

2. Expresar, en notación $O(\cdot)$, el orden que tendría un algoritmo cuyo tiempo de ejecución fuera f(n):

$$f_1(n) = n^2 \qquad f_2(n) = n^2 + 1000n$$

$$f_3(n) = \begin{cases} n & si\ n\ es\ par \\ n^3 & si\ n\ es\ impar \end{cases} \qquad f_4(n) = \begin{cases} n & si\ n \leq 100 \\ n^3 & si\ n > 100 \end{cases}$$

$$f_5(n) = (n-1)^3 \qquad f_6(n) = \sqrt{n^2 - 1}$$

$$f_7(n) = \log(n!) \qquad f_8(n) = n!$$

3. Usando la notación O(·), obtener el tiempo de ejecución de las siguientes funciones:

```
1:  void ejemplo1 (int n)
2:  {
3:    int i, j, k;
4:
5:    for (i = 0; i < n; i++)
6:        for (j = 0; j < n; j++)
7:        {
8:            C[i][j] = 0;
9:            for (k = 0; k < n; k++)
10:               C[i][j] += A[j][k] * B[k][j];
11:       }
12: }
```

```
1:  long ejemplo2 (int n)
2:  {
3:    int  i, j, k;
4:    long total = 0;
5:
6:    for (i = 1; i < n; i++)
7:        for (j = i+1; j <= n; j++)
8:            for (k = 1; k <= j; k++)
9:                total += k*i;
10:
11:   return total;
12: }
```

```
1:  void ejemplo3 (int n)
2:  {
3:    int i, j, x=0, y=0;
4:
5:    for (i = 1; i <= n; i++)
6:        if (i % 2 == 1)
7:        {
8:           for (j = i; j <= n; j++)
9:               x++;
10:          for (j = 0; j < i; j++)
11:              y++;
12:       }
13: }
```

```
1:  int ejemplo4 (int n)
2:  {
3:    if (n <= 1)
4:       return 1;
5:    else
6:       return (ejemplo4(n - 1) + ejemplo4(n - 1));
7:  }
```

```
1:  int ejemplo5 (int n)
2:  {
3:    if (n == 1)
4:       return n;
5:    else
6:       return (ejemplo5(n/2) + 1);
7:  }
```

4. Resolver las siguientes recurrencias:

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

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

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

d. $T(n) = \begin{cases} 0 & si\ n = 0 \\ 1 & si\ n = 1 \\ 3T(n-1)+4T(n-2) & en\ otro\ caso \end{cases}$

e. $T(n) = \begin{cases} 0 & si\ n = 0 \\ 1 & si\ n = 1 \\ 5T(n-1)-8T(n-2)+4T(n-3) & en\ otro\ caso \end{cases}$

f. $T(n) = \begin{cases} 0 & si\ n = 0 \\ 36 & si\ n = 1 \\ 5T(n-1)+6T(n-2)+4\cdot 3^n & en\ otro\ caso \end{cases}$

g. $T(n) = 2T(n-1)+3^n$

h. $T(n) = 2T(n-1)+n+2^n$

i. $T(n) = 2T\left(\frac{n}{2}\right)+\log n$

j. $T(n) = 4T\left(\frac{n}{2}\right)+n$

k. $T(n) = 4T\left(\frac{n}{2}\right)+n^2$

l. $T(n) = 2T\left(\frac{n}{2}\right)+n\log n$

m. $T(n) = \begin{cases} 1 & si\ n = 2 \\ 2T(\sqrt{n})+\log n & si\ n \geq 4 \end{cases}$

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

o. $T(n) = \begin{cases} 1 & si\ n = 1 \\ 5T\left(\frac{n}{2}\right)+(n\log n)^2 & si\ n \geq 2 \end{cases}$

p. $T(n) = \sqrt{n}T(\sqrt{n}) + n,\ \ n \geq 4$

q. $T(n) = nT^2\left(\frac{n}{2}\right),\ \ n > 1,\ \ T(1) = 6$

r. $T(n) = T\left(\frac{n}{2}\right) \cdot T^2\left(\frac{n}{4}\right),\ n \geq 4,\ \ T(1) = 1,\ \ T(2) = 4$

5. El tiempo de ejecución de un algoritmo A viene descrito por la recurrencia

$$T(n) = 7T(n/2) + n^2$$

Otro algoritmo B tiene un tiempo de ejecución dado por

$$T'(n) = aT'(n/4) + n^2$$

¿Cuál es el mayor valor de la constante a que hace al algoritmo B asintóticamente más eficiente que A?

6. Resuelva la siguiente recurrencia:

$$T(n) = aT\left(\frac{n}{b}\right) + n^k$$

con $a \geq 1$, $b \geq 2$ y $k \geq 0$.