

1.- Usando la **notación O**, determinar la eficiencia de las siguientes funciones:

(a)

```
void eficiencia1(int n)
{
    int x=0; int i,j,k; O(1)
    for(i=1; i<=n; i+=4)
        for(j=1; j<=n; j+=[n/4])
            for(k=1; k<=n; k*=2)
                x++; O(1)
}
```

$\left[O(\log_2 n) \right] O(\log_2 n) \left[\begin{array}{l} \text{R. producto} \\ O(n \cdot \log_2 n) \end{array} \right]$

(b)

```
int eficiencia2 (bool existe)
{
    int sum2=0; int k,j,n; O(1)
    if (existe) O(1)
        for(k=1; k<=n; k*=2)
            for(j=1; j<=k; j++)
                sum2++; O(1)
    else
        for(k=1; k<=n; k*=2)
            for(j=1; j<=n; j++)
                sum2++; O(1)
    return sum2;
}
```

$\left[O(k) \right] O(n-1) \left[\begin{array}{l} \text{Regla del peor caso} \\ O(n \cdot \log_2 n) \end{array} \right]$

$\left[O(n) \right] O(n \cdot \log_2 n)$

(c)

<pre>void eficiencia3 (int n) { int j; int i=1; int x=0; O(1) do{ j=1; O(1) while (j <= n){ j=j*2; O(1) x++; O(1) } i++; O(1) }while (i<=n); O(1) }</pre>	<div style="border-left: 1px dashed black; height: 150px; margin: 0 auto; width: 2px;"></div>	<pre>void eficiencia4 (int n) { int j; int i=2; int x=0; O(1) do{ j=1; O(1) while (j <= i){ O(1) j=j*2; O(1) x++; O(1) } i++; O(1) }while (i<=n); O(1) }</pre>
---	---	--

$\left[O(\log_2 n) \right] \left[O(\log_2 i) \right] O(\log_2 n!)$

$O(n \cdot \log_2 n)$

2.- Considerar el siguiente segmento de código con el que se pretende buscar un entero **x** en una lista de enteros L de tamaño n (el bucle **for** se ejecuta **n veces**):

```
void eliminar (Lista L, int x)
{
    int aux, p;
    for (p=primero(L); p!=fin(L);)
    {
        aux=elemento (p,L);
        if (aux==x)
            borrar (p,L);
        else p++;
    }
}
```

Analizar la eficiencia de la función eliminar si:

(a) primero es $O(1)$ y fin, elemento y borrar son $O(n)$. ¿Cómo mejorarías esa eficiencia con un solo cambio en el código?

(b) primero, elemento y borrar son $O(1)$ y fin es $O(n)$. ¿Cómo mejorarías esa eficiencia con un solo cambio en el código?

(c) todas las funciones son $O(1)$. ¿Puede en ese caso mejorarse la eficiencia con un solo cambio en el código?

Consideraciones:

1.- El reto es **individual**

2.- la solución deberá entregarse obligatoriamente en un fichero pdf (se sugiere como nombre reto1.pdf)

3.- Si la solución es correcta, se puntuará con 0.2 para la evaluación continua

4.- El plazo límite de entrega es el 3 de Octubre a las 23.55h

a) $O(n^3)$. Crear variable `int fin = fin(L)`; antes del bucle for, y luego en la condición del for poner `p != fin`.

b) $O(n^2)$. Igual que en el apartado a).

c) $O(n)$. Se podría mejorar si fuera un array de enteros ordenado.