

Antonio Rodríguez Rodríguez. 2º GDADE

Jose Antonio Camara Molina.

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)  $O(n)$ 
        for(j=1; j<=n; j+=[n/4])  $O(n)$ 
            for(k=1; k<=n; k*=2)  $O(\log_2(n))$ 
                x++;  $O(1)$ 
}
```

$O(n^2 \log_2(n))$

(b)

```
int eficiencia2 (bool existe)
{
    int sum2=0; int k,j,n;  $O(1)$ 
    if (existe)
        for(k=1; k<=n; k*=2)  $O(\log_2(n))$ 
            for(j=1; j<=k; j++)  $O(n)$ 
                sum2++;  $O(1)$ 
    else
        for(k=1; k<=n; k*=2)  $O(\log_2(n))$ 
            for(j=1; j<=n; j++)  $O(n)$ 
                sum2++;  $O(1)$ 
    return sum2;  $O(1)$ 
}
```

$O(n \log_2(n))$

(c)

<pre>void eficiencia3 (int n) { int j; int i=1; int x=0; $O(1)$ do{ j=1; while (j <= n){ $O(\log_2(n))$ j=j*2; $O(1)$ x++; $O(1)$ } i++; $O(1)$ }while (i<=n); $O(n)$ }</pre>	<pre>void eficiencia4 (int n) { int j; int i=2; int x=0; $O(1)$ do{ j=1; $O(1)$ while (j <= i){ $O(\log_2(n))$ j=j*2; $O(1)$ x++; $O(1)$ } i++; $O(1)$ }while (i<=n); $O(n)$ }</pre>
---	---

$O(n \log_2(n))$

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):

a)

```
void eliminar (Lista L, int x)
{
    int aux, p;
    for (p=primero(L); p!=fin(L);)  $O(n^2)$ 
    {
        aux=elemento (p,L);  $O(n)$ 
        if (aux==x)
            borrar (p,L);  $O(n)$ 
        else p++;  $O(1)$ 
    }
}
```

$O(n^4)$

Crearía una función "ultimo" en la clase Lista que devolviera el tamaño o número de elementos con eficiencia $O(1)$, sustituyendo $p!=fin(L)$ por $p!=ultimo(L)$.

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 ligero cambio en el código?
- (b) primero, elemento y borrar son $O(1)$ y fin es $O(n)$. ¿Cómo mejorarías esa eficiencia con un ligero cambio en el código?
- (c) todas las funciones son $O(1)$. ¿Puede en ese caso mejorarse la eficiencia con un ligero 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

b)

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

$O(n^2)$

Volvería a hacer lo mismo que en el apartado a, con la función "ultimo" que he inventado. Quedando una eficiencia $O(n)$

c) En este caso no puede mejorarse la eficiencia, quedando esta como $O(n)$. Solo podríamos cambiar la búsqueda lineal por búsqueda binaria, quedando la eficiencia como $\log_2 n$, pero sería un cambio algo más complejo.