

Estructura de Datos  
Act-1.3.2 Notación Asintotica (Funciones Recursivas)

Nombre: David Alejandro Lozano Arreola

Matricula: A01722728

1) ¿Cuál es el orden de cada uno de los siguientes algoritmos?

a) //Entrada: Un entero positivo (n)

```
int algo(int n){  
    if (n <= 0)  
        return 400;  
    else  
        return algo(n-2)+algo(n-2)+algo(n-2)+algo(n-2);  
}
```

$O(4^{n/2})$

b) //Entrada: Un entero positivo (n)

```
int algo(int n){  
    if (n <= 0)  
        return 123;  
    else  
        return algo(n-4)+algo(n-4)+algo(n-4);  
}
```

$O(2^{n/4})$

c) //Entrada: Un entero positivo (n)

```
int algo(int n){  
    if (n == 0)  
        return 400;  
    else  
        return algo(n/2)+algo(n/2)+algo(n/2)+algo(n/2);  
}
```

$O(4^{\log_2(n)})$

d) //Entrada: Un entero positivo (n)

```
int algo(int n){  
    if (n == 0)  
        return 123;  
    else  
        return algo(n/4)+algo(n/4)+algo(n/4);  
}
```

$O(3^{\log_4(n)})$

2) ¿Cuál es la formula recursiva del tiempo de ejecución cada uno de los siguientes algoritmos?

a) //Entrada: Un entero positivo (n)

```
int algo(int n){
    if (n <= 0)
        return 400;
    else
        return algo(n-2)+algo(n-2)+algo(n-2)+algo(n-2);
}
```

$T(n) > 4 * T(n-2) \dots$   
 $T(n) > 4 * 4 * T(n-4) \dots$   
 $\dots$   
 $T(n) > 4 * 4 * 4 * \dots * 4 * T(1)$   
 $n/2$  veces

b) //Entrada: Un entero positivo (n)

```
int algo(int n){
    if (n <= 0)
        return 123;
    else
        return algo(n-4)+algo(n-4)+algo(n-4);
}
```

$T(n) > 3 * T(n-4) \dots$   
 $T(n) > 3 * 3 * T(n-8) \dots$   
 $\dots$   
 $T(n) > 3 * 3 * 3 * \dots * 3 * T(1)$   
 $n/4$  veces

c) //Entrada: Un entero positivo (n)

```
int algo(int n){
    if (n == 1)
        return 400;
    else
        return algo(n/2)+algo(n/2)+algo(n/2)+algo(n/2);
}
```

$T(n) > 4 * T(n/2) \dots$   
 $T(n) > 4 * 4 * T(n/4) \dots$   
 $\dots$   
 $T(n) > 4 * 4 * 4 * \dots * 4 * T(1)$   
 $\log_2(n)$  veces

d) //Entrada: Un entero positivo (n)

```
int algo(int n){
    if (n == 1)
        return 123;
    else
        return algo(n/4)+algo(n/4)+algo(n/4);
}
```

$T(n) > 3 * T(n/4) \dots$   
 $T(n) > 3 * 3 * T(n/8) \dots$   
 $\dots$   
 $T(n) > 3 * 3 * 3 * \dots * 3 * T(1)$   
 $\log_4(n)$  veces