

TC1031: Programación de Estructura de Datos y Algoritmos
Fundamentales Act 1.1.3 Notación Asintótica (Funciones Recursivas)

Ing. Luis Humberto González G.

Forma de Trabajo: Individual.

Nombre: _____

Matricula: _____

1) (20 puntos) ¿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);  
}
```

_____ $O(4^n)$

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);  
}
```

_____ $O(3^n)$

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);  
}
```

_____ $O(4^n)$

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);  
}
```

_____ $O(3^n)$

2) (20 puntos) ¿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);
}
```

$O(4^n)$

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);
}
```

$O(3^n)$

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);
}
```

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

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);
}
```

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

3) (20 puntos) Encuentra la fórmula cerrada de las siguientes fórmulas recursivas:

a)
$$T(n) = \begin{cases} 1, & n = 0 \\ 1 + 5T(n-1), & n > 0 \end{cases}$$

```
int a(int n){
    if (n==0)
        return 1;
    return a(n-1)+a(n-1)+a(n-1)+a(n-1)+a(n-1);
}
```

$O(5^n)$

b)
$$T(n) = \begin{cases} 1, & n = 0 \\ 1 + 8T(n-1), & n > 0 \end{cases}$$

```
int a(int n){
    if (n==0)
        return 1;
    return a(n-1)+a(n-1)+a(n-1)+a(n-1)+a(n-1)+a(n-1)+a(n-1)+a(n-1);
}
```

$O(8^n)$

c)
$$T(n) = \begin{cases} 1, & n = 1 \\ 1 + 4T(n/2), & n > 1 \end{cases}$$

```
int a(int n){
    if(n==1)
        return 123*45;
    return a(n/2)+a(n/2)+a(n/2)+a(n/2);
}
```

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

d) $T(n) = 1, n = 1$
 $T(n) = 1 + 5T(n/3), n > 0$

```
int a(int n){  
    if(n==1)  
        return 123*45;  
    return a(n/3)+a(n/3)+a(n/3)+a(n/3)+a(n/3);  
}
```

$O(n^{\log_3 5})$