

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); }
```

$O(n^2)$

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); }
```

$O(n^{1.585})$

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);  
}
```

1

$$1+4T(n-2)$$

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);
}
```

$$1$$

$$1+3T(n-4)$$

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);
}
```

$$1$$

$$1+4T(n/2)$$

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);
}
```

$$1$$

$$1+3T(n/4)$$

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

$$1, n = 0$$

a) $T(n) = 1 + 5T(n-1), n > 0$

$$5^n$$

$$1, n = 0$$

b) $T(n) = 1 + 8T(n-1), n > 0$

$$8^n$$

$$\begin{aligned} & 1, n = 1 \\ \text{c) } T(n) &= 1 + 4T(n/2), n > 0 \end{aligned}$$

$$N^2$$

$$\begin{aligned} & 1, n = 1 \\ \text{d) } T(n) &= 1 + 5T(n/3), n > 0 \end{aligned}$$

$$N^{1.465}$$