Trabajo Obligatorio

Tiempos y eficiencia.

Leo Joaquin Bruno - FAI 3268

Maria Elvira Monserrat Vidal - FAI 1829

Jeremias Ezequiel Herrera - FAI 3297

05 de Junio del 2022

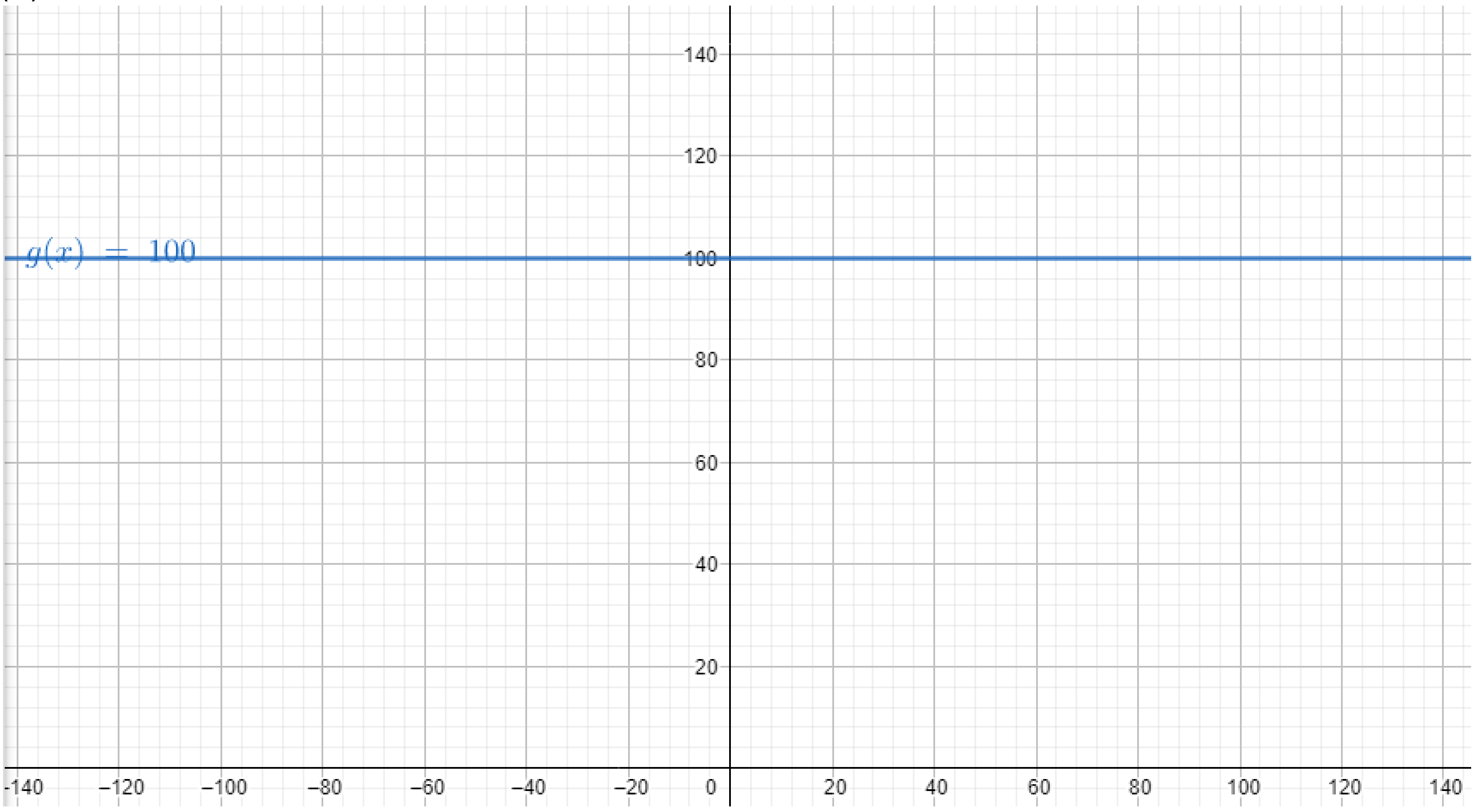
Cálculo de orden. Dadas las siguientes funciones:

a) Representarlas gráficamente utilizando algún graficador.

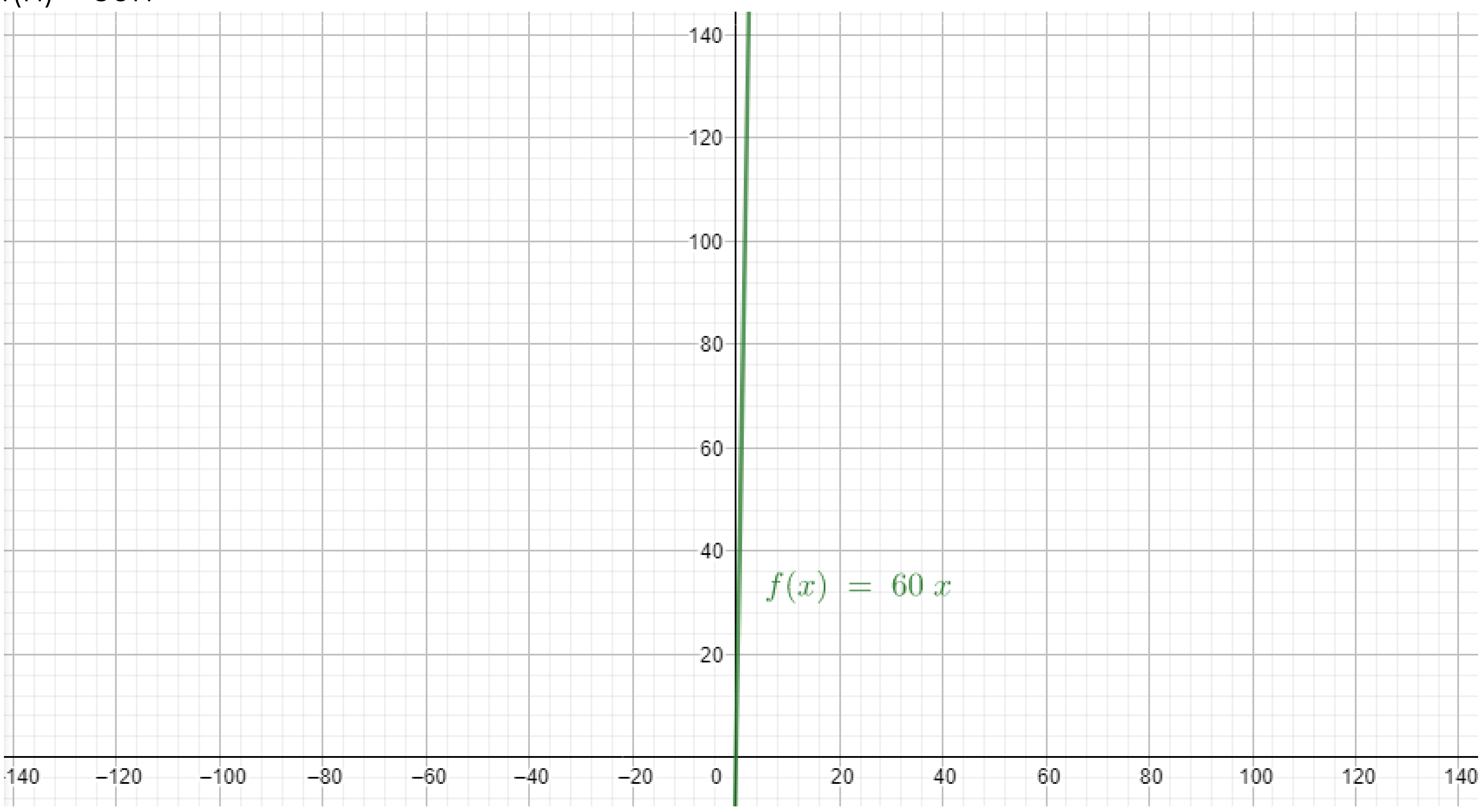
b) Ordenarlas en forma decreciente de acuerdo a su velocidad de crecimiento.

c) Indicar a qué Orden pertenece cada una.

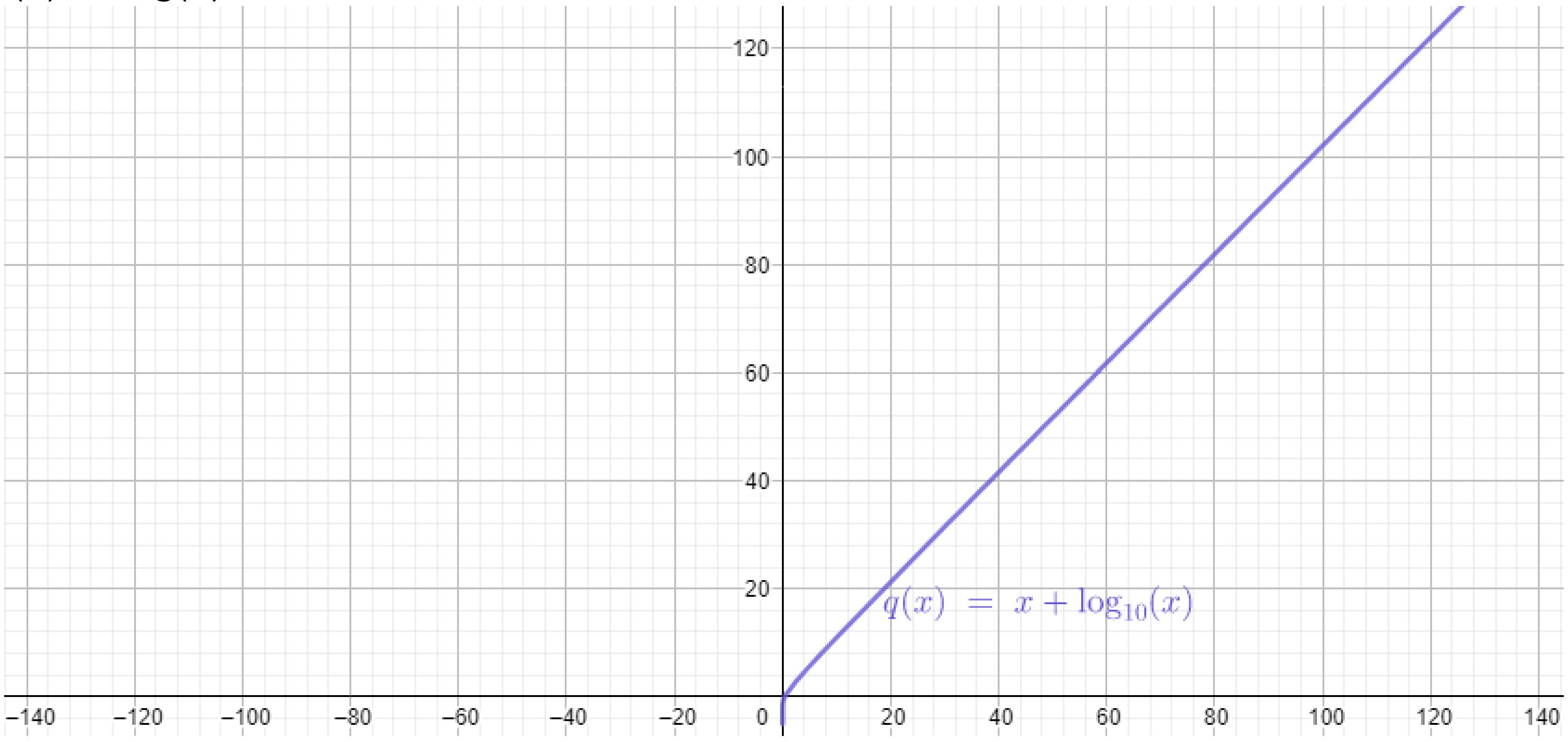
Complejidad constante.
f(n)=100



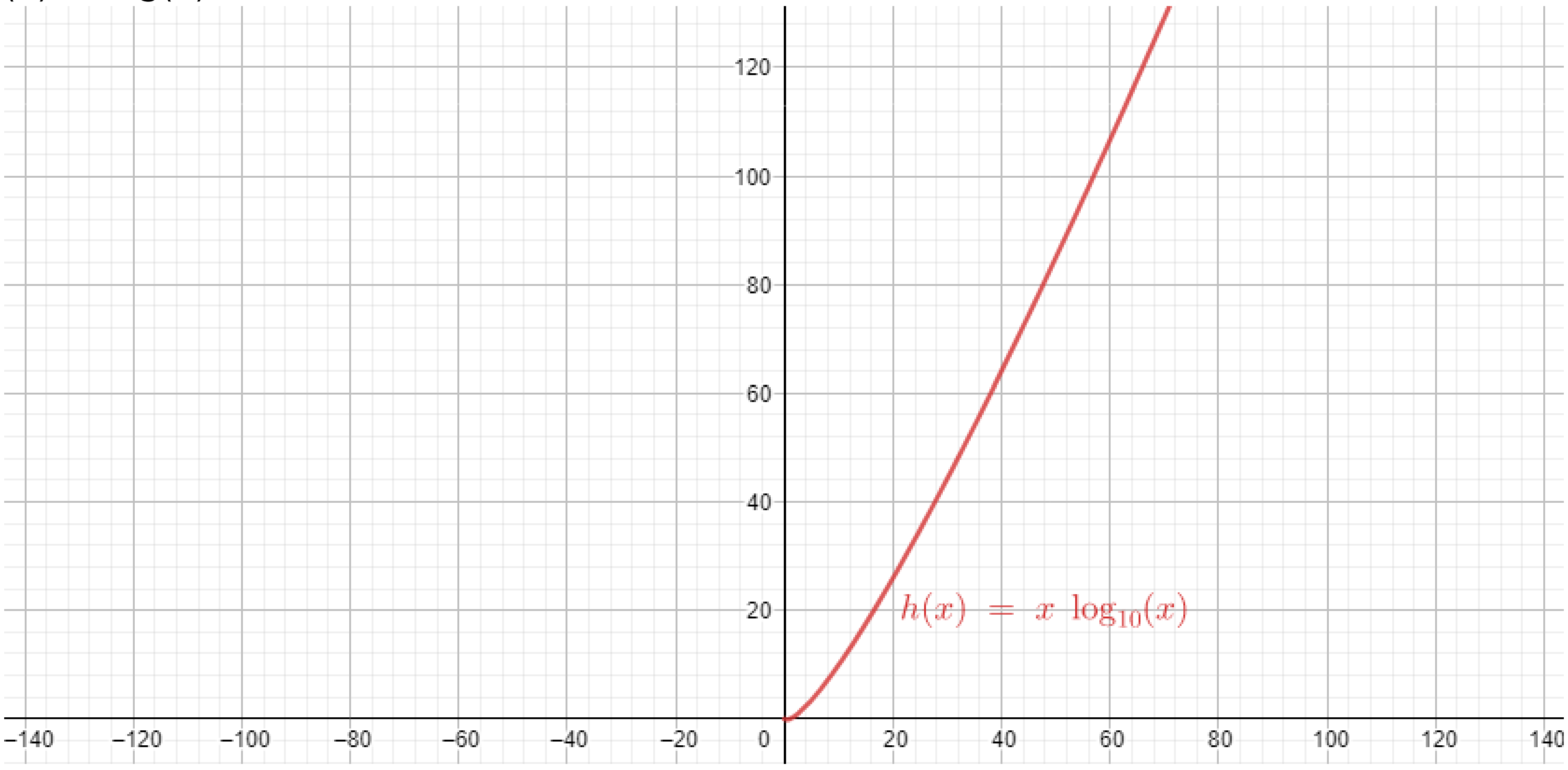
Complejidad lineal.
f(n) = 60n



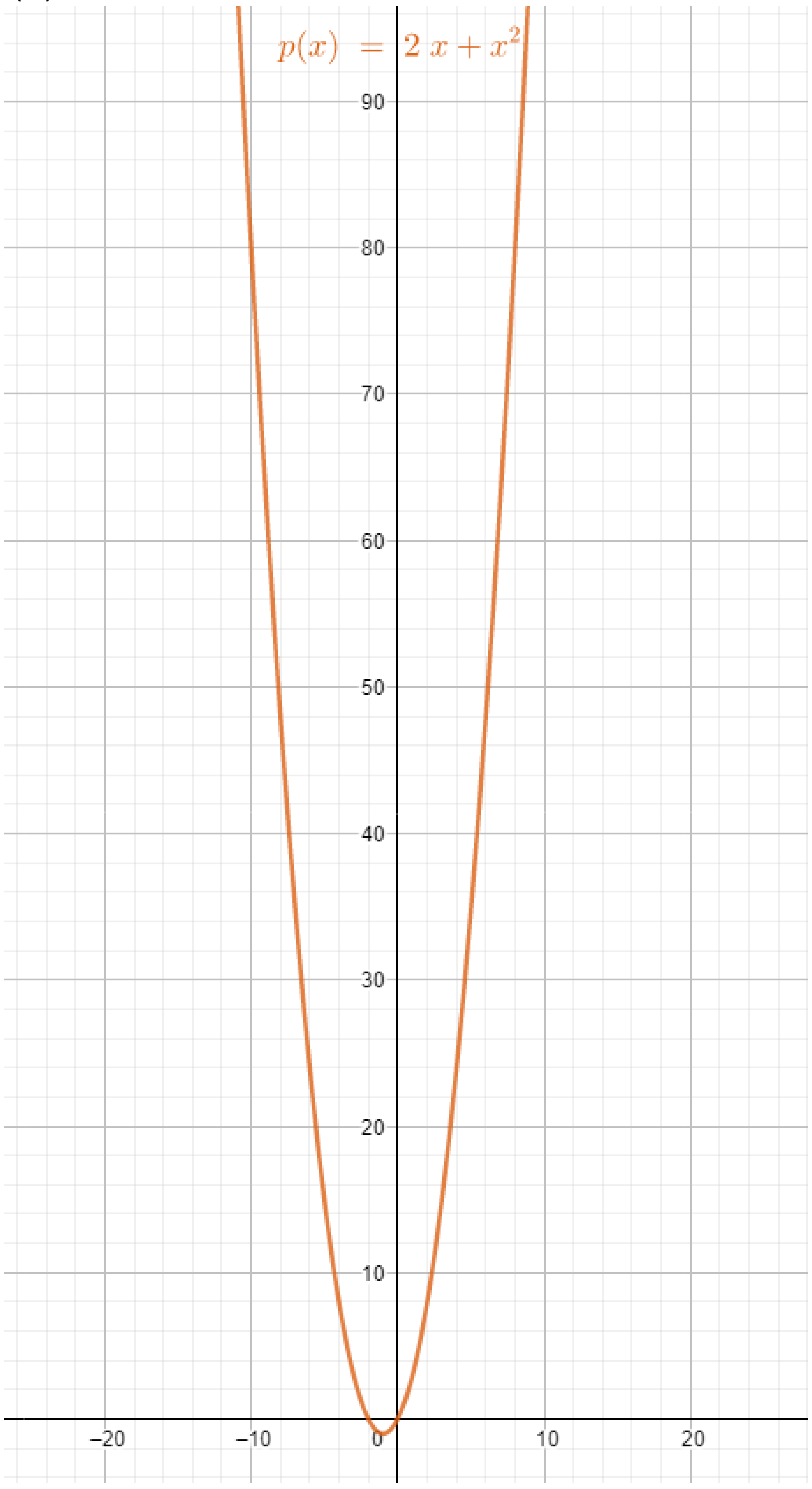
Complejidad lineal.
f(n)=n+log(n)



Complejidad O(n log n).
f(n)=n*log(n)



Complejidad cuadrática.
f(n)=2n+n^2



Corrección:
f(n)= 100 → Orden 1
f(n)= n + log n → orden n
f(n)= 60n → orden n
f(n)= n. log n → orden log
f(n)= 2n+n^2 → orden n^2

Calculando tiempos de ejecución.

Para cada uno de los siguientes subprogramas o segmentos de código:

- ❖ Expresar en función de n el tiempo de ejecución.
- ❖ Establecer el orden utilizando las reglas de la suma y del producto.

tn= t2 + t3 + t5
tn= 1 + 38 + 3 = 42

a)

```
    public static void multiplo()
1   {
2       int n=1;
3       while (!(n == 10))
4           n = n+1;
5       System.out.println("Se iteraron: "+n+ " veces ");
6   }
```

T2 = 1
T3 = 9 * (2 + 2) + 2
T4 = 2
T5 = 3
Tiempo Final = 44

*Suponemos que el SOUT está fuera de la estructura while.
f(n)= 44
Complejidad constante.

b)

```
    Algoritmo Ejemplo
1       Leer (numero)
2       Para i<-1 hasta N con paso 1 Hacer
3           Si (numero mod i=0) entonces
4               Escribir( i,'Es divisor')
5           Fin si
6       Fin para
7   Fin Algoritmo Ejemplo
```

$$T_N = T_1 + T_2$$

$$T_1 = 1$$

$$T_2 = 1 + \sum_{i=1}^{N} (T_3 + 1 + 2) + 1$$

$$T_3 = 2 + T_4 = 4 \qquad T_4 = 2$$

$$T_2 = 2 + \sum_{i=1}^{N} (7) = 2 + 7N$$

$$T_N = 1 + 7N + 2 = 7N + 3$$

f(n)= 7n + 3
Complejidad lineal.

c)

```
  public static void mostrar(int[] arreglo)
1 {
2       int largo = arreglo.length;
3       for (int i=0; i<largo; i++)
4       {
5           System.out.print(arreglo[i]+" - ");
6       }
7 }
```

$$T_N = T_2 + T_3$$

$$T_2 = 2$$

$$T_3 = 1 + \sum_{i=0}^{N-1} (T_5 + 1 + 2) + 1$$

$$T_5 = 3$$

$$T_3 = 2 + \sum_{i=0}^{N-1} (6) = 2 + 6(N-1) = 2 + 6N - 6 = 6N - 4$$

$$T_N = 2 + 6N - 4 = 6N - 2$$

f(n) = 6n - 2
Complejidad lineal.

d)

```
        Algoritmo Ejemplo2
1           Leer (letra)
2           Si (letra='A' OR letra='E' OR letra='I' OR letra='O'OR letra='U')
3               Si (letra='A') entonces
4                   Escribir( 'Es la vocal A')
5               SINO
6                   Si (letra='E') entonces
7                       Escribir( 'Es la vocal E')
8                   SINO
9                       Escribir( 'Es otra vocal')
10                  FIN SI
11              FIN SI
12          SINO
13              Escribir( 'No es vocal')
14          FIN SI
15      Fin Algoritmo Ejemplo2
```

$$T_N = T_1 + T_2$$

$$T_1 = 1$$

$$T_2 = 9 + \max(T_3, T_{13}) = 12$$

$$T_3 = 1 + \max(T_4, T_5) = 3$$

$$T_5 = 1 + \max(T_7, T_9) = 2$$

$T_7 =$ $\quad T_9 = 1 \quad T_4 = 1 \quad T_{13} = 1$

$$T_N = 1 + 12 = 13$$

f(n)= 13
Complejidad constante.

e)

```
        public static boolean verifica(int[] arreglo, int elemento)
1       {
2           int largo = arreglo.length;
3           int i=0;
4           boolean esta=false;
5           while ( i<largo && !esta)
6           {
7               if (arreglo[i]==elemento)
8                   esta= true;
9               i++;
10          }
11          return esta;
12      }
```

$T_N = T_2 + T_3 + T_4 + T_5 + T_{11}$

$T_2 = 2 \quad T_3 = 1 \quad T_4 = 1 \quad T_{11} = 1$

$T_5 = N \cdot (3 + (T_7 + 2)) + 3$

$T_7 = 2 + T_8 = 3 \qquad T_8 = 1$

$T_5 = N \cdot (3 + (3+2)) + 3 = 8N + 3$

$T_N = 2 + 1 + 1 + 8N + 3 + 1 = 8N + 8$

f(n) = 8n + 8
Complejidad lineal.

f)

```
        public static void reemplaza(int[] arreglo,int num, int nuevo)
1       {
2           int largo = arreglo.length;
3           for (int i=0; i<largo; i++)
4           {
5               if (arreglo[i]==num)
6               arreglo[i]=nuevo;
6           }
7       }
```

$T_N = T_2 + T_3$

$T_2 = 2$

$T_3 = 1 + \sum_{i=0}^{N-1} (T_5 + 1 + 2) + 1$

$T_5 = 2 + T_6 = 4 \qquad T_6 = 2$

$T_3 = 2 + \sum_{i=0}^{N-1} (7) = 2 + 7(N-1) = 7N - 5$

$T_N = 2 + 7N - 5 = 7N - 3$

f(n) = 7n - 3
Complejidad lineal.

Tn= T2+T3+T8= 1 + $5n^2$ -3 +2 = $5n^2$

g)

```
        Módulo Misterio(Entero n) retorna vacío
1           Entero i,j,k,s
2           S <-0
3           Para i<- hasta n-1 con paso 1 Hacer
4               Para j<- 1 hasta n con paso 1 Hacer
5                   s<-s+2
6               Fin Para
7           Fin Para
8           ESCRIBIR("s:"+s)
9       Fin Modulo Misterio
```

$T_N = T_2 + T_3 = 1 + 5N^2 - 3 = 5N^2 - 2$

$T_2 = 1$

$T_3 = 1 + \sum_{i=0}^{N-1} (T_4 + 1 + 2) + 1$

$T_4 = 1 + \sum_{j=1}^{N} (T_5 + 1 + 2) + 1 = 2 + \sum_{j=1}^{N} (5) = 2 + 5N$

$T_5 = 2$

$T_3 = 1 + \sum_{i=0}^{N-1} (T_4 + 1 + 2) + 1 = 2 + \sum_{i=0}^{N-1} (5 + 5N) =$

$2 + \sum_{i=0}^{N-1} 5 + \sum_{i=0}^{N-1} 5N = 2 + 5N - 5 + 5N^2 - 5N =$

$5N^2 - 3$

f(n) = $5n^2$

Complejidad cuadrática.

h)

```java
public static void mostrar(int[][] matriz)
{
    for (int f=0; f<matriz.length;f++)
    {
        for (int col=0; col<matriz[0].length; col++)
        {
                System.out.print( matriz[f][col]+" ");
        }
        System.out.println("");
    }
}
```

$$T_N = T_2$$

$$T_2 = 1 + \sum_{F=0}^{N-1}(T_4 + T_8 + 2 + 2) + 2$$

$$T_4 = 1 + \sum_{C=0}^{K-1}(T_6 + 4 + 2) + 3 = 4 + \sum_{C=0}^{K-1}(10) = 10K - 6$$

$$T_6 = 4 \qquad T_8 = 1$$

$$T_2 = 1 + \sum_{F=0}^{N-1}(T_4 + T_8 + 2 + 2) + 2 = 3 + \sum_{F=0}^{N-1}(10K - 6 + 5) =$$

$$3 + \sum_{F=0}^{N-1}(10K - 1) = 3 + (10K-1)(N-1) = 10KN - 10K - N + 1$$

f(n) = 10kn -10k -n +1
Complejidad lineal.

i)

```java
private static void misterio (int n)
{
    int i, j, k=0;
    for (i = 1; i <= n-1; i++)
    {
        for (j = 1; j < i; j++)
        {
            k = i * j;
        }
    }
}
```

$$T_N = T_2 + T_3 = 1 + 2 + \frac{5N^2 - 3N}{2} = 3 + \frac{5N^2 - 3N}{2}$$

$$T_2 = 1$$

$$T_3 = 1 + \sum_{i=1}^{N-1}(T_5 + 2 + 2) + 2$$

$$T_5 = 1 - \sum_{J=1}^{i-1}(T_7 + 1 + 2) + 1 = 2 + \sum_{J=1}^{i-1}(5) = 2 + 5i - 5 = 5i - 3$$

$$T_7 = 2$$

$$T_3 = 1 + \sum_{i=1}^{N-1}(T_5 + 2 + 2) + 2 = 3 + \sum_{i=1}^{N-1}(5i - 3 + 2 + 2) = 3 + \sum_{i=1}^{N-1}(5i + 1) = 3 + \sum_{i=1}^{N-1}(1) + \sum_{i=1}^{N-1}(5i) =$$

$$3 + N - 1 + 5 \cdot \sum_{i=1}^{N-1}(i) = 3 + N - 1 + 5 \cdot \frac{(N-1)((N-1)+1)}{2} = 3 + N - 1 + 5 \cdot \frac{N^2 - N}{2} =$$

$$2 + \frac{2N + 5N^2 - 5N}{2} = 2 + \frac{5N^2 - 3N}{2}$$

f(n)= $5/2\, n^2 - 3/2\, n + 3$
Complejidad cuadrática.

j)

```
    private static int calculo(int n)
1   {
2     int i, j, cantA = 0, cantB = 0;
3     for(i=1; i<=n; i++)
4     {
5         for(j=1; j<i; j++)
6         {
7             cantA++;
8             cantB++;
9         }
10    }
11    return cantB;
12  }
```

$$T_N = T_2+T_3+T_{11} = 2+2+\frac{3N+7N^2}{2}+1 = 5+\frac{7N^2+3N}{2}$$

$$T_2 = 2 \qquad T_{11} = 1$$

$$T_3 = 1+\sum_{i=1}^{N}(T_5+1+2)+1$$

$$T_5 = 1+\sum_{J=1}^{i-1}(T_7+T_8+1+2)+1 = 2+\sum_{J=1}^{i-1} \quad) = 2+7i-7 = 7i-5$$

$$T_7 = 2 \qquad T_8 = 2$$

$$T_3 = 1+\sum_{i=1}^{N}(T_5+1+2)+1 = 2+\sum_{i=1}^{N}\left(7i+(-2)\right) = 2+\sum_{i=1}^{N}(-2)+7\sum_{i=1}^{N}(i) =$$

$$2-2N+7\cdot\frac{N(N+1)}{2} = 2-2N+7\cdot\frac{N^2+N}{2} = 2-2N+\frac{7N^2+7N}{2} =$$

$$2+\frac{-4N+7N^2+7N}{2} = 2+\frac{3N+7N^2}{2}$$

f(n) = $7/2\, n^2 + 3/2\, n + 5$
Complejidad cuadrática.

k)

```
    private static int queHace(int[] a)
1   {
2       //n es la dimensión del arreglo a
3       int n, max, cont, i, j; n = a.length; max = 0;
4       for(i=0; i<n; i++)
5       {
6           cont = 1;
7           j = i + 1;
8           while ( ( j < n ) && ( a[i] <= a[j] ) )
9           {
10              j++;
11              cont++;
12          }
13          if (cont > max)
14              max= cont;
15      }
16      return max;
17  }
```

$$T_N = T_3+T_4+T_{16} = 3+9N^2+4N-11+1 = 9N^2+4N-7$$

$$T_3 = 3 \qquad T_{16} = 1$$

$$T_4 = 1+\sum_{i=0}^{N-1}(T_6+T_7+T_8+T_{13}+1+2)+1$$

$$T_6 = 1 \qquad T_7 = 2$$

$$T_8 = N\cdot\left(5\left(T_{10}+T_{11}\right)\right)+5 = N\cdot(5+4)+5 = 9N+5$$

$$T_{10} = 2 \qquad T_{11} = 2$$

$$T_{13} = 1+T_{14} = 2 \qquad T_{14} = 1$$

$$T_4 = 1+\sum_{i=0}^{N-1}(T_6+T_7+T_8+T_{13}+1+2)+1 = 2+\sum_{i=0}^{N-1}(1+2+9N+5+2+1+2) =$$

$$2+\sum_{i=0}^{N-1}(9N+13) = 2+(9N+13)\cdot(N-1) = 2+9N^2-9N+13N-13 = 9N^2+4N-11$$

f(n) = $9n^2 + 4n - 7$
Complejidad cuadrática.

l)

```
1       j=1;
2       while (j <= n)
3       {
4           if (a[j] <= a[n] )
5               a [n]= a[j];
6           j=j*2;
7       }
```

$$T_N = T_1+T_2 = 1+9N+1 = 9N+2$$

$$T_1 = 1$$

$$T_2 = N\cdot\left(1+\left(T_4+T_6\right)\right)+1$$

$$T_4 = 3+T_5 = 6 \qquad T_5 = 3$$

$$T_6 = 2$$

$$T_2 = N\cdot\left(1+(6+2)\right)+1 = 9N+1$$

f(n)= 9n + 2
Complejidad
Corrección.

T2= (log n)(1+(T4 + T6)) + 1 = 9 log n +1

Tn= 1 + 9 (log n) +1= 9log(n) +2
orden logarítmico.