①

LECCION 4. continuacion
EFICIENCIA de Ordenacion
: SELECCION, INSERCCION
Y BURBUJA.

SELECCION

```
void Intercambiar (int &a, int &b)
{
        int aux = a; ——— O(1)
        a=b; —— O(1)                    Regla del maximo
        b=aux; —— O(1)                        O(1)
}
void  Ordenacion-Seleccion (int *v, int n)
{
   for (int i=0; i<n-1; i++)
     for (int j= i+1; j<n; j++)
        if (v[j] < v[i])          ]—O(1)    n-1
          Intercambiar(v[i], v[j])          Σ   1
                                            j=i+1
}
```

$$\sum_{i=0}^{n-2}\sum_{j=i+1}^{n-1} 1 = \sum_{i=0}^{n-2} (n-1)-(i+1)+1 =$$

$$= \sum_{i=0}^{n-2} n-i-1 = \sum_{i=0}^{n-2} n - \sum_{i=0}^{n-2} i - \sum_{i=0}^{n-2} n \overset{(*)}{=}$$

- $\sum_{i=0}^{n-2} n = n\sum_{i=0}^{n-2} 1 = n\underbrace{(1+1+\dots+1)}_{(n-2)+1} = n(n-1) = n^2-n$

- $\sum_{i=0}^{n-2} i = \underbrace{0+1+2+\dots+n-2}_{\text{PROGRESION ARITMETICA}} = (n-2)\frac{(n-1)}{2} = \frac{n^2}{2}-\frac{n}{2}-n+1 = \frac{n^2}{2}-\frac{3n}{2}+1$

- $\sum_{i=0}^{n-2} 1 = n-1$

$$(*) = (n^2-n) - \left(\frac{n^2}{2} - \frac{3n}{2} + 1\right) - (n-1) = \frac{n^2}{2} + \frac{3n}{2} \boxed{\in O(n^2)}$$

LECCION 3:- continuación

INSERCCIÓN

```
void Ordenacion_Insercion (int *v, int n){
    for (int i=1; i<n; i++)
    {
          int pos=i;  — O(1)
          for (int j=i-1; j≥0; j--)  — $\sum_{j=0}^{i-1} 1$
          {    if (v[j] > v[pos])  ]— O(1)
                    pos=j;
          }
          Intercambiar (v[i],v[pos]); — O(1)
    }
}
```

$$\sum_{i=1}^{n-1}\sum_{j=0}^{i-1} 1 = \sum_{i=1}^{n-1} \underbrace{1+1+\dots+1}_{i} = \sum_{i=1}^{n-1} i$$

$$= \underbrace{1+2+3+\dots+n-1}_{\text{Progresión Aritmética}} = (n-1)\left(\frac{n}{2}\right) = \frac{n^2}{2}-\frac{n}{2} \in \boxed{O(n^2)}$$

LECCION 3: continuación

BURBUJA

```
void Ordenacion_Burbuja(int * v, int n){
    for (int i=1; i<n; i++)
        for (int j=0; j<n-i; j++)
            if (v[j]>v[j+1])
                Intercambiar(v[j], v[j+1]]
}
```

$O(1)$

$\sum_{j=0}^{n-i} 1$

$$\sum_{i=1}^{n-1}\sum_{j=0}^{n-i} 1 = \sum_{i=1}^{n-1} n-i+1$$

$$= \sum_{i=1}^{n-1} n - \sum_{i=1}^{n-1} i + \sum_{i=1}^{n-1} 1 = n(n-1) - \left(n\frac{(n-1)}{2}\right) + (n-1)$$

$$= n^2 - n - \frac{n}{2} + \frac{n}{2} + n - 1 = n^2 - 1 \in \boxed{O(n^2)}$$

- Los tres métodos de ordenación: SELECCION, INSERCCION Y BURBUJA tienen un orden de eficiencia en el peor caso $\boxed{O(n^2)}$

LECCION 4: EFICIENCIA

Ejemplo: bucle anidado condicionado por un if

```
void funcion(int n){
  int x=0, y=0;
  for (int i=1; i<n+1; i++)
    if ((i%2)==0){                      — O(1)
       for (int j=i; j<n+1; j++)   ]— Σ_{j=i}^{n} 1   (1)
          x++;  — O(1)
       for (int j=1; j<i+1; j++)   ]— Σ_{j=1}^{i} 1   (2)
          y++;  — O(1)
    }
  }
}
```

$$\boxed{(1)}\quad \sum_{j=i}^{n} 1 = \underbrace{1+1+\dots+1}_{n-i+1} = n-i+1$$

$$\boxed{(2)}\quad \sum_{j=1}^{i} 1 = 1+1+\dots+1 = i$$

$$O(n-i+1+i) = O(n+1)$$

La condicion hace que se ejecute el cuerpo de i n/2 ya que solamente tenemos n/2 nº pares en n numeros consecutivos.

$$\sum_{i=1}^{n/2} (n+1) = (n+1)\cdot n/2 \in \boxed{O(n^2)}$$

⑥

LECCION 4.- continuacion

Ejemplo: El bucle es controlado por una condicion. Se usa bucle while.

```
void funcion(int n){
    int x=2, contador=0;
    while (x<=n){
        x *=2;
        contador++;
    }
    cout <<contador;
}
```

$x *= 2;\ contador++; \to O(1)$

¿Cuantas veces se cumple la condicion?

Ej. $n=8$ $\underbrace{x=2,\ x=4\ \ x=8}_{3veces}$

$n=2$ $\underbrace{x=2}_{1vez}$

$n=16$ $\underbrace{x=2,\ x=4,\ x=8\ \ x=16}_{4veces}$

Claramente se hace $\log_2(n)$ veces.

$$\sum_{contador=0}^{\log_2(n)} 1 = \log_2(n)+1 \in \boxed{O(\log_2(n))}$$

Ejemplo: Tres bucles - MULTIPLICACION de MATRICES.

```
void funcion(int n){
    for (int i=0; i<n; i++)
        for (int j=0; j<n; j++){
            C[i][j]=0;
            for (int K=0; K<n; K++)
                C[i][j]+= A[i][K] * B[K][j];
    }
```

$O(n^3)$

$O(n)$

$O(n^2)$

3

$$\sum_{i=0}^{n-1}\sum_{j=0}^{n-1}\sum_{K=0}^{n-1} 1 = n^3 \in \boxed{O(n3)}$$

LECCION 4: continuación.

Ejemplo: Tres bucles con diferentes límites

```
void funcion (int n){
    int suma=0;
    for (int i=1; i<n; i++)
        for (int j=i+1; j≤n; j++)
            for (int k=1; k<j+1; k++)
                suma += i+j+k;  ← O(1)
```

$\sum_{k=1}^{j} 1$ $\qquad \sum_{j=i+1}^{n}$

$$\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}\sum_{k=1}^{j}1 = \sum_{i=1}^{n-1}\sum_{j=i+1}^{n} j = \sum_{i=1}^{n-1}(i+1)+(i+2)+\dots+n =$$

$$= \sum_{i=1}^{n-1}(n-i)\frac{(n+i+1)}{2} = \sum_{i=1}^{n-1}\frac{n^2-i^2+n-i}{2} = \frac{1}{2}\left(\sum_{i=1}^{n-1}n^2 - \sum_{i=1}^{n-1}i^2 + \sum_{i=1}^{n-1}n - \sum_{i=1}^{n-1}i\right)$$

$$\sum_{i=1}^{n-1} n^2 = n^2(n-1) = n^3-n^2$$

$$\sum_{i=1}^{n-1} i^2 = \frac{(n-1)\,(n)\,(2n-1)}{6} = \frac{1}{6}\left(2n^3-2n^2-n^2+n\right) = \frac{1}{6}\left(2n^3-3n^2+n\right)$$

$$\sum_{i=1}^{n-1} n = n(n-1) = n^2-n$$

$$\sum_{i=1}^{n-1} i = 1+2+\dots+n-1 = \frac{(n-1)}{2}(n) = \frac{n^2-n}{2}$$

Juntando todos los términos

$$\frac{1}{2}\left(n^3-n^2-\frac{1}{6}\left(2n^3-3n^2+n\right)+(n^2-n)\cancel{+}-\left(\frac{n^2-n}{2}\right)\right)=$$

$$\frac{1}{2}\left(\frac{2}{3}n^3-\frac{2}{3}n\right) \in \boxed{O(n3}$$

⑧ LECCION 4.- continuacion

Ej: Examen Sep 2012

a)

```
int n,j; int x=0;
[illegible]; int i=1;
do {
    j=1
    while (j≤n) {
        j=j*2;
        x++;
    }
    i++;
}while (i<n);
```

$\sum_{j=1}^{\log_2(n)}$

$$\sum_{i=1}^{n}\sum_{j=1}^{\log_2(n)} 1 = \sum_{i=1}^{n}\log_2(n) = n\log_2(n)$$

$O(n\cdot\log_2(n))$

b)

```
int n,j; int i=2; int x=0;
do{
    j=1;
    while (j≤i){
        j=j*2;
        x++;
    }
    i++;
}while (i<=n)
```

$\sum_{j=1}^{\log_2(i)} 1$

$$\sum_{i=2}^{n}\sum_{j=1}^{\log_2(i)} 1 = \sum_{i=2}^{n}\log_2(i) =$$

$\log_2(2)+\log_2(3)+....+\log_2(n)$

$= \log(2\cdot3\cdot4\cdot....\cdot n) = \underline{\log_2(n!)}$

$O(\log_2(n!)) \le O(n\cdot\log_2(n))$

Ej: Examen Julio 2006

a)

```
int sum1=0; int K,j,n;
for (K=1; K≤n; K*=2)
    for (j=1; j≤n; j++)
        sum1++;
```

$\sum_{j=1}^{n} 1 = n$

$\sum_{K=1}^{\log_2(n)} n = n\cdot\log_2(n) \in \boxed{O(n\cdot\log_2(n))}$

b)

```
int sum2=0; int K,j,n;
for (K=1; K≤n; K*=2)
    for (j=1; j≤K; j++)
        sum2++;
```

~~Σ~~

$\sum_{j=1}^{K} 1 = K$

$\sum_{K=1}^{\log_2(n)} K = 1+2+...+\log_2(n)$

$= \log_2(n)\cdot\frac{(\log_2(n)+1)}{2}$

$= \frac{\log_2(n)\log_2(n)+\log_2(n)}{2}$

$\in O(\log_2(n)\cdot\log_2(n))$

⑨

LECCION 4

Ej: Examen Julio 2006 a

Ordenar de ~~mayor~~ menor a mayor

- $\log_2(2n\cdot\log_2(n)) \rightarrow \log_2(2n)+\log_2(\log_2(n)) = 1+\log_2(n)+\log_2(\log_2(n)) \in O(\log_2(n))$
- $n\cdot\log_2(\sqrt{n}) \rightarrow n\cdot\log_2(n^{1/2}) = \frac{n}{2}\log_2(n) \in O(n\cdot\log_2(n))$
- $n\sqrt{n} \rightarrow n\cdot n^{1/2} = n^{3/2} \in O(n^{3/2})$
- $2^{\log_2(n)} \rightarrow n \in O(n)$
- $(1.00001)^n \rightarrow O(1)$
- $2^{2\log_2 n}$ — $(2^{\log_2(n)})^2 = n^2 \in O(n^2)$
- $n^2\cdot 2^{3\log_2(n)}$ — $n^2\cdot(2^{\log_2(n)})^3 = n^2\cdot n^3 = n^5 \in O(n^5)$

Ahora viendo los ordenes de eficiencia a los que pertenecen ordenamos

$$(1.00001)^n \leq \log_2(2n\cdot\log_2(n)) \leq 2^{\log_2(n)} \leq n\log_2(\sqrt{n}) \leq n\cdot\sqrt{n}$$
$$\leq 2^{2\log_2(n)} \leq n^2\cdot 2^{3\log_2(n)}$$

Ej: Examen Junio 2008

Ordenar de menor a mayor

- $\sqrt{n}$ — $n^{1/2} \in O(n^{1/2})$
- $n^3+1 \in O(n^3)$
- $\left(\frac{n^4}{n^2+1}\right) \in O(n^2)$
- $n\cdot\log_2(n^2) \in 2\cdot n\log_2(n) \in O(n\cdot\log_2(n))$
- $n\cdot\log_2\log_2(n^2) = n\cdot\log_2(2\cdot\log_2(n)) = n\cdot\log_2(2)+n\cdot\log_2\log_2(n) = \in O(n\log_2\log_2(n))$
- $3^{\log_2(n)} = 3^{\frac{\log_3(n)}{\log_3(2)}} = (3^{\log_3(n)})^{1/\log_3(2)} = n^{1/\log_3(2)} \simeq n \in O(n)$
- 3^n — $\in O(3^n)$
- $2^{100} \in O(1)$
- $n+100 \in O(n)$

$$2^{100} \leq \sqrt{n} \leq n+100 \overset{\leftrightarrow}{\leq} 3^{\log_2(n)} \leq n\log_2\log_2(n^2) \leq n\log_2(n^2)$$
$$\leq \frac{n^4}{n^2+1} \leq n^3+1$$

LECCION 4:
EFICIENCIA FUNCIONES RECURSIVAS

Ej1:

```
int factorial(int n){   <----- TIEMPO de EFICIENCIA T(n)
    if (n ≤ 1)   <-----┐  COMPARACION O(1) siempre la hacemos
        return 1;      ┘— O(1) si entramos en el if
    else
      return n*factorial(n-1);
}
```

$\underbrace{n*factorial(n-1)}_{T(n-1)+1}$

$+1$ por la devolucion y producto.

TIEMPO de EFICIENCIA para el factorial

$$T(n)=\begin{cases} 1+T(n-1) & \text{para } n>1 \text{ o } n\geq 2 \\ 1 & \text{para } n\leq 1 \end{cases}$$

RESOLVEMOS

$$T(n) = 1 + T(n-1) \qquad n\geq 2$$

$$\hookrightarrow 1+T(n-2) \qquad n\geq 3$$

$$T(n) = 1+1+T(n-2) = 2+T(n-2) \qquad n\geq 3$$

$$\hookrightarrow 1+T(n-3) \qquad n\geq 4$$

$$T(n) = 3 + T(n-3) \qquad n\geq 4$$

$\vdots$

$$T(n) = k + T(n-k) \qquad n\geq k+1$$

$\vdots$

cuando $k=n-1$ $\qquad n\geq 1$

$$T(n)=(n-1)+T(n-(n-1))$$

$$T(n)=(n-1)+T(1) \qquad (\text{donde } T(1)=1 \text{ caso base})$$

$$T(n)=(n-1)+1 \in \boxed{O(n)}$$

LECCION 4.- FUNCIONES RECURSIVAS

(fin - inicio) + 1 = n

Ej 2: Búsqueda Binaria

```
int   BB ( int *v, int inicio, int fin, int x) { ← T(n)
        if (inicio ≤ fin) {
            int m = (fin + inicio)/2;  ← O(1)
            if (v[m] == x)    ← O(1)
                return m;
            else {
              if (v[m] > x)  ← O(1)
                 return BB(v, inicio, m-1, x); ← T(n/2)
              else
                 return BB (v, m+1, fin, x); ← T(n/2)

            }

        }
        else
            return -1;

    }
```

$$T(n) = \begin{cases} 1 + T(n/2) & \text{para } n \geq 2 \\ 1 & \text{para } n \leq 1 \end{cases}$$

$T(n) = 1 + T\left(\frac{n}{2}\right) \quad n \geq 2$

Hacemos cambio de variable

$n = 2^m \Rightarrow \log_2(n) = m$

$n \geq 2 \Rightarrow m \geq 1$

$T(2^m) = 1 + T(2^{m-1}) \quad m \geq 1$

$\quad\quad\quad\quad\quad\quad\quad \llcorner 1 + T(2^{m-2}) \quad m \geq 2$

$T(2^m) = 2 + T(2^{m-2}) \quad m \geq 2$

$\vdots$

$T(2^m) = K + T(2^{m-K}) \quad m \geq K$

$\vdots$

$T(2^m) = m + T(2^{m-m})$ para $K = m$.

$T(2^m) = m + T(1) = m + 1 \Longrightarrow T(n) = \log_2(n) + 1 \in \boxed{O(\log_2(n))}$

①

$$T(n) = \begin{cases} n + T\left(\frac{n}{2}\right) & n \geq 2 \quad m \geq 1 \\ 1 & n = 1 \implies m = 0 \end{cases}$$

[redacted] ~~T(1)~~

$$T(n) = n + T\left(\frac{n}{2}\right) \qquad n \geq 2$$

Hacemos cambio de variable $n = 2^m$ $m = \log_2(n)$

$$T(2^m) = 2^m + T\left(\frac{2^m}{2}\right) \qquad m \geq 1$$

$$T(2^m) = 2^m + T(2^{m-1}) \qquad m \geq 1$$

$$\hookrightarrow 2^{m-1} + T(2^{m-2})$$

$$T(2^m) = 2^m + 2^{m-1} + T(2^{m-2}) \qquad m \geq 2$$

$$\llcorner 2^{m-2} + T(2^{m-3}) \quad m \geq 3$$

T

$$T(2^m) = 2^m + 2^{m-1} + 2^{m-2} + T(2^{m-3}) \qquad m \geq 3$$

Para $K \leq m$

$$T(2^m) = 2^m + 2^{m-1} + 2^{m-2} + \ldots + 2^{m-K+1} + T(2^{m-K}) \quad m \geq K$$

Para $K = m$

$$T(2^m) = 2^m + 2^{m-1} + \ldots + 2^{m-m+1} + T(0) \qquad m = K$$

$$= 2^m + 2^{m-1} + \ldots + 2^1 + 1$$

$$= \frac{2^{m+1} - 1}{2 - 1} \in \cancel{O(2} = 2^{m+1} - 1$$ deshacemos el cambio de variable

$$2 \cdot n - 1 \in O(n)$$

③

$$T(n) = \begin{cases} T(n/2) + n^2 & n \geq 2 \\ 1 & n = 1 \end{cases}$$

Cambio de variable
$n = 2^m$

$$T(2^m) = \begin{cases} T(2^{m-1}) + (2^m)^2 & m \geq 1 \\ 1 & m = 0 \end{cases}$$

$$T(2^m) = T(2^{m-1}) + 2^{2m} = T(2^{m-1}) + 4^m \quad m \geq 1$$

$$\llcorner T(2^{m-2}) + 4^{(m-1)} \quad m \geq 2$$

$$T(2^m) = T(2^{m-2}) + 4^{(m-1)} + 4^m \quad m \geq 2$$

$$\llcorner T(2^{m-3}) + 4^{(m-2)}$$

$$T(2^m) = T(2^{m-3}) + 4^{(m-2)} + 4^{(m-1)} + 4^m \quad m \geq 3$$

Para $k < m$

$$T(2^m) = T(2^{m-k}) + 4^{(m-k+1)} + 4^{(m-k+2)} \dots + 4^m \quad m \geq k$$

Para $k = m$

$$T(2^m) = T(2^0) + 4 + 4^2 + 4^3 + \dots + 4^m$$

$$= 1 + 4 + 4^2 + 4^3 + \dots + 4^m$$

$$T(2^m) = \frac{4^{m+1} - 1}{3} \longrightarrow$$ deshaciendo el cambio de variable

$$= \frac{4^m \cdot 4 - 1}{3} = \frac{(2^m)^2 \cdot 4 - 1}{3} = \frac{4n^2 - 1}{3} \in O(n^2)$$

④

$$T(n) = \begin{cases} 2\,T(\sqrt{n}) + \log_2(n) & n \geq 4 \\ 1 & n = 2 \end{cases}$$

cambio de variable $2^{2^m} = n \Rightarrow \log_2(n) = 2^m \quad \log_2(\log_2(n)) = m.$

$$T(2^{2^m}) = \begin{cases} 2\;T((2^{2^m})^{1/2}) + 2^m & m \geq 1 \\ 1 & m = 0 \end{cases}$$

$$T(2^{2^m}) = 2 \cdot T(2^{2^{m-1}}) + 2^m \qquad m \geq 1$$

$$\llcorner\; 2 \cdot T(2^{2^{m-2}}) + 2^{m-1}$$

$$T(2^{2^m}) = 2^2\, T(2^{2^{m-2}}) + 2 \cdot 2^{m} \qquad m \geq 2$$

$$\llcorner\; 2\,T(2^{2^{m-3}}) + 2^{m-2} \qquad m \geq 3$$

$$T(2^{2^m}) = 2^3\, T(2^{2^{m-3}}) + 3 \cdot 2^{m-2} \qquad m \geq 3$$

Para $k < m$

$$T(2^{2^m}) = 2^k\, T(2^{2^{m-k}}) + k \cdot 2^m \qquad m > k$$

Para $k = m$

$$T(2^{2^m}) = 2^m \cdot T(2^{2^0}) + m \cdot 2^m \qquad m = k$$

$$= \log_2(n) + \log_2 \log_2(n) \cdot \log_2(n)$$

$$\in O(\log_2(n) \cdot \log_2 \log_2(n))$$

⑤

$$T(n) = \begin{cases} T(\sqrt{n}) + \log_2 \log_2 n + \log_2 n & n \geq 4 \\ 1 & n = 2 \end{cases}$$

$n = 2^{2^m}$

$$T(2^{2^m}) = T(2^{2^{m-1}}) + m + 2^m \qquad m \geq 1$$

$$T(2^{2^m}) = T(2^{2^{m-2}}) + m-1 + m + 2^{m-1} + 2^m \quad m \geq 2$$

$$T(2^{2^m}) = T(2^{2^{m-3}}) + m-2 + m-1 + m + 2^{m-2} + 2^{m-1} + 2^m \quad m \geq 3$$

Para $k < m$

$$T(2^{2^m}) = T(2^{2^{m-k}}) + m-k+1 + m-k+2 \ldots + m + 2^{m-k+1} + \ldots 2^m \; m$$

Para $k = m$

$$T(2^{2^m}) = \underset{11}{T(2^{2^0})} + \cancel{m} 1 + 2 + \ldots + m + 2 + 2^2 + \ldots + 2^m \qquad m = k$$

$$= \underbrace{1 + 2 + \ldots m}_{} + 1 + 2 + 2^2 + \ldots + 2^m$$

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

$$= \left(\log_2 \log_2(n) + 1\right) \frac{\log_2 \log_2(n)}{2} + 2 \cdot \log_2(n) - 1$$

$$= \frac{\left(\log_2 \log_2(n)\right)^2}{2} + \frac{\log_2 \log_2(n)}{2} + 2 \log_2(n) - 1$$

$$\in O(\log_2(n))$$

$T(n) = 2T(n/2) + n^3 \quad n \geq 2 \quad T(1) = 1$

$n = 2^m \Rightarrow n^3 = 8^m \quad [(2^m)^3 = 2^{3m} = 2^{3^m} = 8^m] \quad 2^m \geq 2^1 \Leftrightarrow m \geq$

$$\left. \begin{array}{l} T(2^m) = 2\underbrace{T(2^{m-1})}_{\substack{\shortparallel \\ 2T(2^{m-2}) + 8^{m-1}}} + [8^m] \quad m \geq 1 \end{array} \right\} = 2^2 T(2^{m-2}) + 2 \cdot 8^{m-1} + 8$$

$m \geq 2$

$$\left. \begin{array}{l} T(2^m) = 2^2 \underbrace{T(2^{m-2})}_{\substack{\shortparallel \\ 2T(2^{m-3}) + 8^{m-2}}} + [2 \cdot 8^{m-1} + 8^m] \quad m \geq 2 \end{array} \right\} = 2^3 T(2^{m-3}) + 2^2 8^{m-2} + 2^1 8^{m-} + 2^0 8^{m-0}$$

$m \geq 3$

$T(2^m) = 2^3 T(2^{m-3}) + [2^2 \, 8^{m-2} + 2^1 \, 8^{m-1} + 2^0 \, 8^{m-0}] \quad m \geq 3$

En general, para $m \geq i$ tenemos:

$T(2^m) = 2^i T(2^{m-i}) + [2^{i-1} 8^{m-(i-1)} + \dots + 2^2 8^{m-2} + 2^1 8^{m-1} + 2^0 8^{m}]$

y para $i = m$:

$T(2^m) = 2^m \underbrace{T(1)}_{\substack{\shortparallel \\ 1}} + [2^{m-1} \, 8^1 + \dots + 2^2 8^{m-2} + 2^1 8^{m-1} + 2^0 8^m]$

Por tanto:

$T(2^m) = 2^m 8^0 + 2^{m-1} 8^1 + 2^{m-2} 8^2 + \dots + 2^2 8^{m-2} + 2^1 8^{m-1} + 2^0 8^m =$

$= 2^m + 2^{m-1} 2^3 + 2^{m-2} (2^3)^2 + \dots + 2^2 (2^3)^{m-2} + 2 (2^3)^{m-1} + (2^3)^m =$

$= 2^m + 2^{m+2} + 2^{m+4} + \dots + 2^{3m-4} + 2^{3m-2} + 2^{3m} =$

[Progresión geométrica de razón $2^2 = 4$]

$$= \frac{2^{3m} \cdot 4 - 2^m}{4 - 1} = \frac{4}{3}(2^m)^3 - \frac{1}{3} 2^m \quad [2^m = n] =$$

$= \frac{4}{3} n^3 - \frac{1}{3} n \quad \Rightarrow T(n)$ es $\underline{\underline{O(n^3)}}$