(2)

LECCIÓN 4: continuación

EFICIENCIA de Ordenación: SELECCIÓN, INSERCCIÓN Y BURBUJA.

SELECCIÓN

```
void Intercambiar (int &a, int &b)
{
    int aux = a;   —— O(1)
    a = b;         —— O(1)
    b = aux;       —— O(1)
}
```

Regla del máximo $O(1)$

```
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)
          Intercambiar(v[i], v[j])
}
```

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

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

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

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

- $\sum_{i=0}^{n-2} i = \underbrace{0+1+2+\dots+n-2}_{\text{PROGRESIÓN ARITMÉTICA}} = (n-2)\dfrac{(n-1)}{2} = \dfrac{n^2}{2}-\dfrac{n}{2}-n+1 = \dfrac{n^2}{2}-\dfrac{3n}{2}+1$

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

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

③ LECCIÓN 4

INSERCIÓN

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

$O(1)$ ← int value=v[i];

$\sum_{j=0}^{i-1} 1$ ← for

$O(1)$ ← Intercambiar(v[i], v[j+1]);

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

$$= \underbrace{1+2+3+\dots+n-1}_{\text{Progresion aritmética.}} = (n-1)\cdot\frac{n}{2} = \frac{n^2}{2} - \frac{n}{2} \in O(n^2)$$

(4)

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]
}
```

$O(1)$ $\qquad \sum_{j=0}^{n-i} 1$

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

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

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

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

5

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.- continuación

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

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

$-O(1)$

¿Cuantas veces se cumple la condición?

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

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

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

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

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

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];
    }
```

$O(n^3)$ $O(n^2)$ $O(n)$

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

7

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)
```

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

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

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

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

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

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

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

Juntando todos los términos

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

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

8

LECCION 4.- continuación

Ej: Examen Sep 2012

a)

```
int n,j; int x=0;
; 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)
```

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

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

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

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

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

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++;
```

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

b)

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

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

$$\sum_{k=1}^{\log_2(n)} k = 1+2+\dots+\log_2(n)$$

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

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

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

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

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

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