

Estructura de Datos

Act-1.3.1 Notación Asintotica(Funciones Iterativas)

Nombre: Alejandro Barrera Bejarano

Matricula: A01254672

1) *Contesta las preguntas en base al siguiente algoritmo*

```
s = 0
for (int i=1; i<=n; i++)
    s = s + i * i
return s
```

- a) **¿Qué realiza el algoritmo?** calcula la suma de los cuadrados de los números enteros desde 1 hasta n
- b) **¿Cuál es la operación básica?** la multiplicación de un número por sí mismo
- c) **¿Cuántas veces se realiza la op. básica?** n veces
- d) **¿Cuál es el orden del algoritmo?** $O(n)$

2) *¿Cuál es el orden de cada uno de los siguientes algoritmos?*

a) **// Entrada: Matriz A[0..n-1, 0..n-1] de números reales.**

```
for (int i=0; i<= n-2; i++)
    for (int j=i+1; j<n; j++)
        for (int k=i; k<n; k++)
            A[i,k] = A[j,k] - A[i,k] * A[j,i] / A[i,i]       $O(n^3)$ 
```

b) **//Entrada: Un entero positivo (n)**

```
int Q(int n){
    if (n==1)
        return 1
    return n;
}       $O(1)$ 
```

c) **//Entrada: Un entero positivo (n)**

```
int P(int n){
    int acum = 0;
    if (n==0)
        return 0
    else
        if (n % 2 == 0)
            for (int i=1; i<n; i*=2)
                acum += i;
    else
        return n;
}       $O(\log n)$ 
```

d) //Entrada: Un entero positivo (n)

```
int a=0;
int b=n;
for (int i=1; i<= 2*n; i++) {
    a++;
    b+=a;
    c*=(a+b);
}
b=c+a;
```

_____ $O(n)$

e) //Entrada: Un entero positivo (n)

```
int acum=1;
for (int i=1; i<=n; i++)
    for (int j=i; j<=n; j++)
        acum+=(i*j);
```

_____ $O(n^2)$

f) //Entrada: Un entero positivo (n)

```
int b=1;
j = n;
while (j>=0) {
    b++;
    j--;
```

_____ $O(n)$

g) //Entrada: Un entero positivo (n)

```
int acum=1;
for (int i=1; i<=n; i+=2)
    for (int j=i; j<=n; j++)
        acum+=(i*j);
```

_____ $O(n^2)$

h) //Entrada: Un entero positivo (n)

```
int acum=1;
for (int i=1; i<=n; i*=2)
    for (int j=i; j<=n; j+=2)
        acum+=(i*j);
```

_____ $O(n \log n)$

Estructura de Datos
Act-1.3.2 Notación Asintotica (Funciones Recursivas)

Nombre: Alejandro Barrera Bejarano

Matricula: A01254672

1) ¿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);  
}
```

$O(2^n)$

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);  
}
```

$O(3^{(n/4)})$

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);  
}
```

$O(\log n)$

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);  
}
```

$O(\log n)$

2) ¿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);
}
```

$$T(n) = 4T(n-2) + c$$

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);
}
```

$$T(n) = 3T(n-4) + c$$

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);
}
```

$$T(n) = 4T(n/2) + c$$

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);
}
```

$$T(n) = 3T(n/4) + c$$