

TC1031: Programación de Estructura de Datos y Algoritmos
Fundamentales Act 1.1.2 Notación Asintótica (Funciones Iterativas)

Ing. Luis Humberto González G.

Forma de Trabajo: Individual.

Nombre: _____

Matricula: _____

1) (10 puntos) 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 números enteros al cuadrado hasta llegar a 'n'
- b) ¿Cuál es la operación básica? s=s+i*i
- c) ¿Cuántas veces se realiza la op. Básica? 'n' veces
- d) ¿Cuál es el orden del algoritmo? O(n)

2) (40 puntos) ¿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(n²)

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(n²)

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(n²)

3) (50 puntos) Escribe un algoritmo que dado un arreglo que contiene enteros positivos, regrese la suma de los enteros impares contenidos en el arreglo.

a) Realiza el algoritmo en forma iterativa, ¿Cuál es el orden del algoritmo?

```
int sumaPares(int n){
```

```
    if (n<=0)
```

```
        Return 0;
```

```
    else if (n%2==0)
```

```
        return n1+n2+n3...  
    else  
        cout<<"no hay pares"<<endl;  
    }
```

O(n)

- b) Realiza el algoritmo en forma recursiva ¿Cuál es el orden del algoritmo?

```
int sumaPares(int n){  
    if (n<=0)  
        return 0;  
    else if (n%2==0)  
        return n+sumaPares(n-2);  
    else  
        return sumaPares(n-1);  
}
```

O(n)