

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?**

El algoritmo calcula la suma de los cuadrados de los números enteros desde 1 hasta n.

**b) ¿Cuál es la operación básica?**

La operación básica es la multiplicación de cada número (i) por sí mismo (i \* i), seguida de la adición del resultado a la variable acumuladora (s).

**c) ¿Cuántas veces se realiza la op. básica?**

La operación básica se realiza una vez en cada iteración del bucle for. Dado que el bucle ejecuta n veces, la operación básica se realiza n veces en total.

**d) ¿Cuál es el orden del algoritmo?**

El orden del algoritmo es  $O(n)$ , ya que el número de operaciones básicas es directamente proporcional al valor de n (la entrada del algoritmo).

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)**

//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;
}
```

$O(\log n)$  en el peor caso (cuando n es par) y  $O(1)$  en el caso en que n es impar.

**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);
```

$O(n \log n)$