

UNIDAD TEMÁTICA 4

Arboles Binarios y Recursividad

PRÁCTICOS DOMICILIARIOS INDIVIDUALES - FORMULACIÓN DE PSEUDOCÓDIGO

Ejercicio #1

1. Desarrolla en pseudocódigo, en forma recursiva, un algoritmo para calcular el factorial de un cierto número entero que se pasa como parámetro (descrito en el slide #4 de la ppt de clase sobre recursividad).

- Identifica claramente el caso base y la sentencia que lo contempla.
- ¿Puedes verificar que siempre el algoritmo progresará hacia el caso base?

2. Analiza el orden del tiempo de ejecución del algoritmo.

3. Escribe, en lenguaje natural, casos de prueba apropiados para testear este algoritmo

4. Implementa el algoritmo (en JAVA) y pruébalo (mediante test – cases apropiados):

- ¿Qué sucede si el número es negativo?
- Verifica que factorial(4), factorial(5), y factorial(0) produzcan los resultados esperados

```
public static int factorial(int n){ //O(LOG(N))
    if(n==0){ //CASO BASE O(1)
        return 1; //O(1)
    } else{
        return (n*factorial(n-1)); //O(1)
    }
}
```

Si el número es negativo tirara el error "Stack Overflow"

```
System.out.println(factorial(4)); //Resultado: 24
System.out.println(factorial(5)); //Resultado: 120
System.out.println(factorial(0)); //Resultado: 1
```

Ejercicio #2

1. Desarrolla en pseudocódigo, en forma recursiva, el algoritmo Algoritmo SumaLineal(A, n), que se describe en el slide #11 de la ppt de clase sobre recursividad.
 - Identifica claramente el caso base y la sentencia que lo contempla.
 - ¿Puedes verificar que siempre el algoritmo progresará hacia el caso base?
2. Analiza el orden del tiempo de ejecución del algoritmo
3. Escribe, en lenguaje natural, casos de prueba apropiados para testear este algoritmo
4. Implementa en JAVA los test-cases correspondientes al punto 3.
 - ¿Qué sucede si el parámetro n es negativo?
 - ¿Qué sucede si el vector A está vacío?
5. Implementa el algoritmo (en JAVA) y pruébalo con los test-cases. Luego repara eventuales errores.

```
public static int sumaLineal(int[] A, int n){ //O(N)
    if (n==1){ //CASO BASE O(1)
        return A[0]; //O(1)
    } else{
        return (sumaLineal(A, n-1) + A[n-1]); //O(1)
    }
}
```

Si el parámetro n es negativo aparece el problema "Stack Overflow".

Si el vector A está vacío el resultado es 0.