

SOLEMNE 1
ESTRUCTURAS DE DATOS Y ALGORITMOS (CIT-2006) 1^{ER} SEMESTRE 2023

Nombre: _____ Rut: _____

1)	
2)	
3)	
Nota	

- Cada respuesta debe ser justificada con claridad.
- Durante el desarrollo de la prueba no se responde ningún tipo de pregunta.
- Ningún alumno puede salir de la sala durante el desarrollo de la evaluación.

DE ACUERDO AL ARTÍCULO 50 DEL REGLAMENTO DE ESTUDIANTES DE PREGRADO

"Cualquier conducta de un o una estudiante que vicie o tienda a viciar una actividad o evaluación académica, ya sea que se ejecute antes, durante o luego de su realización, dará origen a una o más de las siguientes sanciones, según la gravedad de la falta cometida: nota mínima (1,0) en la respectiva evaluación; reprobación del curso respectivo; suspensión por un semestre o año académico, o expulsión de la Universidad."

Pregunta 1: Algoritmo Misterioso

Usted recibe una carta con un algoritmo misterioso en ella. Dado que usted es un experto en la materia, responderá las siguientes preguntas acerca de este misterioso algoritmo:

```

1  static int evaluatePostfix(String expr) {
2      Stack<Integer> stack = new Stack<>();
3
4      for (int i = 0; i < expr.length(); i++) {
5          char c = expr.charAt(i);
6
7          if (c == ' ') {
8              continue;
9          }
10
11         if (Character.isDigit(c)) {
12             stack.push(c - '0');
13         } else {
14             int v1 = stack.pop();
15             int v2 = stack.pop();
16
17             switch (c) {
18                 case '+':
19                     stack.push(v2 + v1);
20                     break;
21                 case '-':
22                     stack.push(v2 - v1);
23                     break;
24                 case '/':
25                     stack.push(v2 / v1);
26                     break;
27                 case '*':
28                     stack.push(v2 * v1);
29                     break;
30             }
31         }
32     }
33     return stack.pop();
34 }
```

Pista: la expresión $c - '0'$ tiene como resultado el valor del número entero c . Ejemplos: $'0' - '0' = 0$, $'4' - '0' = 4$.

1. Para cada uno de los siguientes casos de prueba, indique el valor retornado por el método `AlgoritmoMisterioso` al recibir el caso como argumento: **(12 puntos)**

- $5\ 5\ +\ =\ 10$

-
- $6\ 6\ /\ 9\ *\ =\ 9$
 - $2\ 4\ *\ 2\ 2\ *\ +\ =\ 8\ +\ 4\ =\ 12$

2. En base a sus observaciones del punto anterior, explique en no más de **3 líneas**, ¿De qué forma el algoritmo modifica los datos que recibe? (**3 puntos**)

El algoritmo recibe como input una expresión aritmética en notación de postfijo (los operadores van después del operando) y retorna el resultado de esta.

3. Analice el algoritmo en cuestión y describa su tiempo de ejecución en términos de $\mathcal{O}(f(n))$, donde n es el largo del arreglo recibido como argumento y f es una función matemática propuesta por usted. Fundamente su respuesta. (**15 puntos**)

El tiempo de ejecución es $O(N)$.

El algoritmo itera en la expresión aritmética de manera lineal, de izquierda a derecha. Realizando operaciones constantes en cada iteración.

Pregunta 2: Suma de dos listas

Se define una lista enlazada de la siguiente forma:

```
1  class LinkedList {
2      class Node {
3          int value;
4          Node next;
5      }
6      Node head;
7  }
```

Considere el uso de esta lista enlazada para representar números enteros: con un dígito por nodo, en orden inverso de los dígitos (el último nodo representa el dígito más significativo). Por ejemplo, la lista 2 -> 9 -> null representa el número 92. La lista 1 -> 3 -> 9 -> null representa el número 931.

Considerando lo anterior se le solicita que:

1. Defina un método `suma`, el cual recibe como argumentos los head de dos listas que representan números y retorna el head de una nueva lista, la cual representa la suma de los dos números recibidos. **(20 puntos)**

Como ejemplo:

```
Caso 1:
Input:
    l1: 8 -> 3 -> 2 -> null
    l2: 3 -> 4 -> 3 -> null
Output:
    suma: 1 -> 8 -> 5 -> null

Caso 2:
Input:
    l1: 7 -> 1 -> 9 -> null
    l2: 5 -> 9 -> null
Output:
    suma: 2 -> 1 -> 0 -> 1 -> null
```

Un ejemplo de implementación puede ser la siguiente:

```
static LinkedList.Node suma(LinkedList.Node n1, LinkedList.Node n2) {
    LinkedList.Node dummyHead = new LinkedList.Node();
    LinkedList.Node current = dummyHead;
    int carry = 0;

    while (n1 != null || n2 != null) {
        int x, y;
        if (n1 == null)
            x = 0;
        else
            x = n1.value;
        if (n2 == null)
            y = 0;
        else
            y = n2.value;

        int sum = carry + x + y;
        carry = sum / 10;
        current.next = new LinkedList.Node();
        current.next.value = sum % 10;
        current = current.next;
        if (n1 != null)
```

```

        n1 = n1.next;
        if (n2 != null)
            n2 = n2.next;
    }

    if (carry > 0) {
        current.next = new LinkedList.Node();
        current.next.value = carry;
    }

    return dummyHead.next;
}

```

2. Analice el algoritmo diseñado y describa su tiempo de ejecución en términos de $\mathcal{O}(f(n, m))$, donde n y m son la cantidad de datos en la primera y segunda lista, respectivamente, y f es una función matemática propuesta por usted. Fundamente su respuesta. (10 puntos)

El algoritmo inicializa una nueva lista y luego itera en ambas listas hasta llegar al final de ambas. En cada iteración:

- Se suman los valores de los nodos actuales con el valor del acarreo.
- Luego, se obtiene el valor del nuevo nodo calculando el modulo 10 (para obtener solo el dígito de unidad del resultado)
- Y se actualiza el valor del acarreo dividiendo por 10 (para obtener el dígito de la decena del resultado).
- Finalmente se agrega el nuevo nodo a la nueva lista.

Si después de recorrer ambas listas todavía hay un acarreo pendiente, se agrega un nodo adicional con ese acarreo.

El tiempo de ejecución es $\mathcal{O}(\max(n, m))$ o $\mathcal{O}(n+m)$.

Pregunta 3: La mejor alternativa

Para cada consulta debe indicar la mejor respuesta.

1. ¿Cuánto puede crecer, como máximo, el tiempo de ejecución de un algoritmo de complejidad cuadrática ($\mathcal{O}(n^2)$) al triplicar la cantidad de datos de entrada?
 - a) **2**
 - b) **3**
 - c) **8**
 - d) **9** [Respuesta correcta]
2. ¿Cuáles son las operaciones básicas de una fila (cola FIFO)?
 - a) Agregar un elemento al final de la fila y sacar el elemento del inicio de la fila. [Respuesta correcta]
 - b) Agregar un elemento al final de la fila y editar el elemento al inicio de la fila.
 - c) Agregar un elemento al inicio de la fila y sacar el elemento del inicio de la fila.
 - d) Agregar un elemento al final de la fila y sacar el elemento del final de la fila.
3. ¿Cuáles son los algoritmos de ordenamiento “in-place” (usan $\mathcal{O}(\log N)$ memoria extra)?
 - a) Insertion-Sort y Merge-Sort.
 - b) Insertion-Sort y Selection-Sort. [Respuesta correcta]
 - c) Selection-Sort y Merge-Sort.
 - d) Selection-Sort.
4. ¿Cuál es la complejidad de Merge-Sort en el peor caso?
 - a) $\mathcal{O}(n \log(n))$ [Respuesta correcta]
 - b) $\mathcal{O}(n(\log(n))^2)$
 - c) $\mathcal{O}(n^2)$
 - d) $\mathcal{O}(n^2 \log(n))$
5. ¿Cuál es la complejidad de Merge-Sort en el mejor caso?
 - a) $\mathcal{O}(n)$
 - b) $\mathcal{O}(n \log(n))$ [Respuesta correcta]
 - c) $\mathcal{O}(n(\log(n))^2)$
 - d) $\mathcal{O}(n^2)$

Esta pregunta sirve de bonus. Cada respuesta correcta vale **1 punto**. El puntaje final de la prueba será el mínimo entre el puntaje total obtenido entre las tres preguntas y **60**.