

# Estructura de datos y algoritmos

Rodrigo Alvarez

[rodrigo.alvarez2@mail.udp.cl](mailto:rodrigo.alvarez2@mail.udp.cl)

# Encuesta

Código ramo: 2006

Sección: 03



## 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`.

## 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`.

Para cada uno de los siguientes casos de prueba, indique el valor retornado por el método al recibir el caso como argumento:

- 5 5 +
- 6 6 / 9 \*
- 2 4 \* 2 2 \* +

## 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`.

Para cada uno de los siguientes casos de prueba, indique el valor retornado por el método al recibir el caso como argumento:

- 5 5 + = 10
- 6 6 / 9 \* = 9
- 2 4 \* 2 2 \* + =12

## 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`.

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?

## 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`.

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?

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

## 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:  $'0' - '0' = 0$ .  $'4' - '0' = 4$ .

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



## 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`.

Analice el algoritmo en cuestión y describa su tiempo de ejecución en términos de  $O(f(n))$ , dónde  $n$  es el largo del arreglo recibido como argumento y  $f$  es una función matemática propuesta por usted. Fundamente su respuesta.

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.

### 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.

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.

```

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;
}

```

### 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.

Analice el algoritmo diseñado y describa su tiempo de ejecución en términos de  $O(f(n, m))$ , 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.

### 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.

Analice el algoritmo diseñado y describa su tiempo de ejecución en términos de  $O(f(n, m))$ , 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.

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  $O(\max(n, m))$ .

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  $O(\log N)$  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)  $\mathcal{O}(n^2)$