

# Estructuras de Datos y Algoritmos: Guía 1

Yerko Ortiz

## 1. Problemas de desarrollo

Resuelva los siguientes problemas diseñando un algoritmo en papel. Pruebe su correctitud y describa el tiempo de ejecución en términos de  $\mathcal{O}(f(n))$ .

### 1. Invertir un entero

Dado un número entero  $N$ , diseñe un algoritmo que invierta sus dígitos.

#### **Ejemplo 1**

Input:  $N = 2674$

Output: 4762

### 2. Arreglo circular

Dado un arreglo  $A$  de números enteros, con tamaño  $N$ . Diseñe un algoritmo que rota el arreglo a la izquierda  $r$  posiciones.

#### **Ejemplo 1**

Input:

$N = 4$

$A = \{5, 7, 8, 9\}$

$r = 1$

Output:  $\{7, 8, 9, 5\}$

#### **Ejemplo 2**

Input:

$N = 4$

$A = \{5, 7, 8, 9\}$

$r = 0$

Output:  $\{5, 7, 8, 9\}$

### 3. Arreglo incremental

Dado un arreglo de  $N$  enteros, se le pide modificar el arreglo de forma que sea incremental, es decir que cada elemento en la secuencia sea igual o mayor que el elemento previo. Para realizar esta modificación en cada movimiento la única operación disponible es incrementar el valor de un elemento cualquiera en el arreglo en 1. Diseñe un algoritmo que retorne la cantidad mínima de modificaciones.

#### **Ejemplo**

Input:

$N = 5$

$A = \{2, 4, 3, 5\}$

Output: 2

### 4. Todos los pares de un arreglo

Sea un arreglo de  $N$  enteros diseñe un algoritmo que retorne una matriz de  $N$  filas y 2 columnas dimensiones que contenga cada par posible en el arreglo.

#### **Ejemplo**

Input:

$N = 3$

$A = \{1, 2, 3\}$

Output:

$\{ \{1, 1\}, \{1, 2\}, \{1, 3\},$

$\{2, 1\}, \{2, 2\}, \{2, 3\},$

$\{3, 1\}, \{3, 2\}, \{3, 3\} \}$

### 5. Multiplicación de matrices

Sean dos matrices  $M_1$  y  $M_2$ , ambas matrices de  $N$  filas y  $N$  columnas. Diseñe un algoritmo que reciba dos matrices de enteros como entrada y retorne la matriz resultante al multiplicar ambas matrices.

#### **Ejemplo** Input:

$N = 2$

$M_1 = \{\{2, 2\}, \{3, 3\}\}$

$M_2 = \{\{5, 5\}, \{10, 10\}\}$

Output:  $\{\{30, 30\}, \{45, 45\}\}$

## 2. Notación asintótica

Caracterize cada uno de los siguientes tiempos de ejecución en función de  $\mathcal{O}(f(N))$ .

1.  $T(N) = 5N^2 + 8N + 7$

2.  $T(N) = 24$

3.  $T(N) = \sqrt{N} + \log_2 N$

4.  $T(N, K) = N \log_2 N + N^K$

5.  $T(N, K) = \log_2 N + N^K, \forall K \geq 1$

6.  $T(N) = N \log_2 N + N \log_2 \log_2 N$

7.  $T(N, M) = 8N^3 + 5M^23 + N^2 + 1$

### 3. Análisis de algoritmos

Para cada uno de los siguientes métodos en Java:

- Describa el problema que resuelven en términos de Input/Output.
- Caracterize el tiempo de ejecución utilizando notación  $\mathcal{O}(f(n))$ .
- Calcule el output resultante para los casos de prueba.

#### 1.

```
1  public static String binary(int N) {
2      if (N == 0) return "0";
3      int lgn = (int) (Math.log(N) / Math.log(2));
4      char[] bits = new char[lgn + 1];
5      do {
6          bits[lgn--] = (char) (N % 2 + '0');
7          N /= 2;
8      } while(N > 0);
9
10     return String.valueOf(bits);
11 }
```

Casos de prueba:

- Input:  $N = 5$
- Input:  $N = 82$
- Input:  $N = 25$

#### 2.

```
1  public static int euclid(int a, int b) {
2      int r;
3      while (b > 0) {
4          r = a % b;
5          a = b;
6          b = r;
7      }
8      return a;
9  }
```

Casos de prueba:

- Input:  $a = 8, b = 4$

- Input:  $a = 9, b = 5$
- Input:  $a = 1232, b = 4$

### 3.

```

1  public static boolean[] eratosthenes(int N) {
2      boolean[] is_prime = new boolean[N + 1];
3      Arrays.fill(is_prime, true);
4      is_prime[0] = is_prime[1] = false;
5
6      for (int i = 2; i <= N; i++) {
7          if (is_prime[i] && i * i <= N) {
8              for (int j = i * i; j <= N; j += i) {
9                  is_prime[j] = false;
10             }
11         }
12     }
13
14     return is_prime;
15 }

```

Casos de prueba:

- Input:  $N = 10$
- Input:  $N = 32$

### 4.

```

1  public static int f1(int N) {
2      return (N*N + N) / 2;
3  }

```

Casos de prueba:

- Input:  $N = 5$
- Input:  $N = 500$

## 5.

```
1 public static int f2(int N) {  
2     int s = 0;  
3     for(int i = 0; i <= N; i++)  
4         s += i;  
5     return s;  
6 }
```

Casos de prueba:

- Input:  $N = 7$
- Input:  $N = 9$

## 6.

```
1 public static int f3(int N) {  
2     int s = 0;  
3     for (int i = 0; i <= N; i++) {  
4         for (int j = 0; j < i; j++) {  
5             s++;  
6         }  
7     }  
8     return s;  
9 }
```

Casos de prueba:

- Input:  $N = 3$
- Input:  $N = 6$