

Objetivos

Unidad 1: Análisis de Algoritmos

La actividad planteada en esta hoja de trabajo contribuye al desarrollo del siguiente objetivo específico:

OE1.4. Analizar algoritmos independientes de una implementación concreta (no dependiente del lenguaje de programación).

OE1.8. Especificar formalmente un problema teniendo en cuenta sus precondiciones y postcondiciones.

1. Enunciado

```
24 import java.io.BufferedReader;
7 public class MainA {
8     public static void main(String[] args)
9         throws IOException{
10         BufferedReader br = new BufferedReader
11             (new InputStreamReader(System.in));
12         BufferedWriter bw = new BufferedWriter
13             (new OutputStreamWriter(System.out));
14
15         String linea = br.readLine();
16         int n = Integer.parseInt(linea);
17         int[] A = new int[n];
18         int k = 0;
19         for (int i = 0; i < n; i++) {
20             linea = br.readLine();
21             int valor = Integer.parseInt(linea);
22             A[i] = valor;
23             if(k<valor) k = valor;
24         }
25         int[] C = new int[k+1];
26         for (int i = 0; i <= k; i++) {
27             C[i] = 0;
28         }
29         for (int i = 0; i < n; i++) {
30             C[A[i]] = C[A[i]] + 1;
31         }
32         for (int i = 1; i <= k; i++) {
33             C[i] = C[i] + C[i-1];
34         }
35         int[] B = new int[n];
36         for (int i = n-1; i >= 0; i--) {
37             B[C[A[i]]-1] = A[i];
38             C[A[i]] = C[A[i]]-1;
39         }
40         for (int i = 0; i < B.length; i++) {
41             bw.write(B[i]+"\\n");
42         }
43         bw.close();
44         br.close();
45     }
46 }
```

```

20 import java.io.BufferedReader;
7  public class MainB {
8      public static class Numero{
9          private Numero sig; private int valor;
10         public Numero(int v){ valor = v; sig = null;}
11         public int darValor(){ return valor;}
12         public Numero darSiguiente(){ return sig;}
13         public void cambiarSiguiente(Numero s){ sig = s;}
14     }
15     public static void main(String[] args)
16         throws IOException{
17         BufferedReader br = new BufferedReader
18             (new InputStreamReader(System.in));
19         BufferedWriter bw = new BufferedWriter
20             (new OutputStreamWriter(System.out));
21
22         String linea = br.readLine();
23         int n = Integer.parseInt(linea);
24         linea = br.readLine();
25         int valor = Integer.parseInt(linea);
26         Numero primero = new Numero(valor);
27         for (int i = 1; i < n; i++) {
28             linea = br.readLine();
29             valor = Integer.parseInt(linea);
30             Numero nuevo = new Numero(valor);
31             if(valor<=primero.darValor()){
32                 nuevo.cambiarSiguiente(primero);
33                 primero = nuevo;
34             }else{
35                 Numero actual = primero;
36                 while(actual.darSiguiente()!=null
37                     && valor>actual.darSiguiente().darValor()){
38                     actual = actual.darSiguiente();
39                 }
40                 nuevo.cambiarSiguiente(actual.darSiguiente());
41                 actual.cambiarSiguiente(nuevo);
42             }
43         }
44         while(primero!=null){
45             bw.write(primero.darValor()+"\n");
46             primero = primero.darSiguiente();
47         }
48         bw.close();
49         br.close();
50     }
51 }

```

Considere la siguiente entrada y salida deseada para los dos algoritmos anteriores y con base en esto responda:

- | | | |
|----------------|---------------|---|
| Entrada | Salida | a. ¿Los dos algoritmos producen la salida indicada ante la entrada dada? Para justificar su respuesta usted debe hacer un seguimiento a cada uno para verificar si esto es así o no. [4pts] Para el algoritmo A, usted debe escribir únicamente el estado del arreglo B al final de cada iteración del ciclo donde es llenado. [4pts] Para el algoritmo B, usted debe escribir únicamente el estado de la lista cuyo primer elemento es la variable primero, al inicio de cada iteración del ciclo for. |
| 10 | 2 | |
| 23 | 8 | |
| 30 | 12 | |
| 21 | 15 | |
| 40 | 18 | |
| 15 | 21 | b. Con base en la entrada dada, y los algoritmos propuestos usted debe inferir el problema. Escriba la especificación del problema (recuerde: [2pts] nombre, [3pts] entradas y [3pts] salida). Los dos algoritmos resuelven el problema que usted especificó para todas las entradas con el mismo formato? Si es así explique brevemente por qué (máximo 4 líneas). |
| 8 | 23 | |
| 12 | 30 | |
| 36 | 36 | |
| 18 | 40 | |

Solución del seguimiento

a. Algoritmo 1: el algoritmo produce la salida esperada ante la entrada dada

→ El algoritmo realiza 11 iteraciones durante el ciclo para llenar el Arreglo B.

Arreglo B

[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]
2	8	10	12	15	18	21	23	30	36	40

Algoritmo 2: el algoritmo también produce la salida esperada ante la entrada.

→ La lista

23, 21, 15, 8, 1, 12, 15, 18, 21, 23, 30, 36, 40

b. Algoritmo 1: Ordenamiento por posición (nombre)

Entrada: lista de #s a ordenar

Salida: lista de #s ordenada

Algoritmo 2: Ordenamiento por entrada (nombre)

Entrada: lista de #s a ordenar

Salida: lista ordenada de #s.