

1. Para cada uno de los tiempos que toma un algoritmo en terminar, hallar el orden de complejidad correspondiente:

a)  $T_A(n) = 2n^3 - 3n + 1$

$$T_A(n) = 2n^3 - 3n + 1 = O(n^3)$$

b)  $T_A(n) = n^5 + 4^2 - \sqrt{n} + 1$

$$T_A(n) = n^5 + 4^2 - \sqrt{n} + 10 = O(n^5)$$

c)  $T_A(n) = n^2 \log n - 2n^3 \log n + 2$

$$T_A(n) = n^2 \log n - 2n^3 \log n + 2 = O(n^3 \log n)$$

2. Calcule la complejidad del siguiente algoritmo

3. void XXXXXX(int n)	=O(1)
4. {	
5.     int x = 0;	=O(1)
6.     for (int i = 1; i <= n; i += 5)	=O(n)
7.     {	
8.         int j = 1;	=O(1)
9.         while (j <= n)	=O(log n)
10.             {	
11.                 x = x + j;	=O(1)
12.                 j *= 4	=O(1)
13.             }	=O(log n)
14.         for (int k = n; k >= 1; k -= 2)	=O(n)
15.             {	
16.                 x = x + 1;	=O(1)
17.             }	=O(n)
18.     }	=O(n*n log n)
19. }	=O(n <sup>2</sup> log n)

20. (10%) Suponga que se tienen los siguientes dos algoritmos para resolver el mismo problema, y suponga que la función **buscar** tiene complejidad  $O(n \log n)$ .

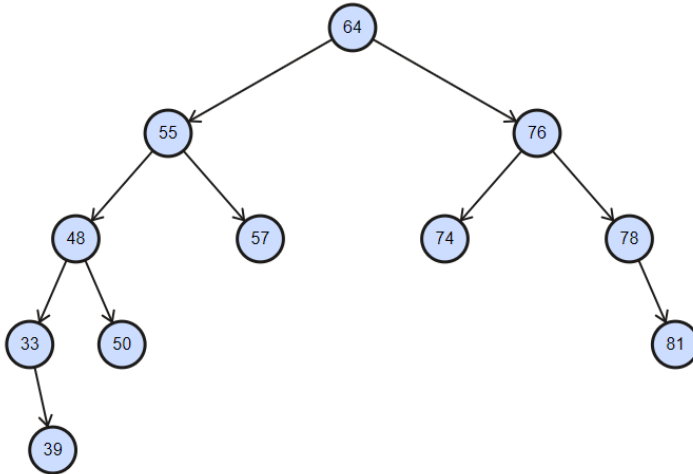
<pre> <b>boolean</b> Ex1(int[] a, int elem) {     <b>int</b> pos = <i>buscar</i>(a, elem);     <b>int</b> n = a.length;     <b>int</b> x = pos;      <b>for</b> (int i = 0; i &lt; n; ++i) {         x += 2;         <b>for</b> (int j = 0; j &lt; n; ++j) {             <b>if</b> (a[j] &gt; a[pos]) {                 x++;             }         }     }     <b>return</b> x &gt; elem; } </pre>	<pre> =0(1) =0(n log n) =0(1) =0(1)  =0(n) =0(1) =0(n) =0(1) =0(1) =0(1) =0(n) =0(n²) =0(1) =0(n²) </pre>
<pre> <b>boolean</b> Ex2(int[] a, int elem) {     <b>int</b> n = a.length;     <b>int</b> x = 0;      <b>for</b> (int i = 0; i &lt; n; ++i) {         <b>int</b> pos = <i>buscar</i>(a, elem);         x += pos + 2;         <b>for</b> (int j = 0; j &lt; n; ++j) {             <b>if</b> (a[j] &gt; a[pos]) {                 x++;             }         }     }     <b>return</b> x &gt; elem; } </pre>	<pre> =0(1) =0(1) =0(1)  =0(n) =0(n log n) =0(1) =0(n) =0(1) =0(1) =0(1) =0(n) =0(n² log n) =0(1) =0(n² log n) </pre>

Indique con cuál de los dos algoritmos se queda para resolver el problema. Justifique muy bien su respuesta.

## Desarrollo de Software

### Ejercicios sobre árboles binarios

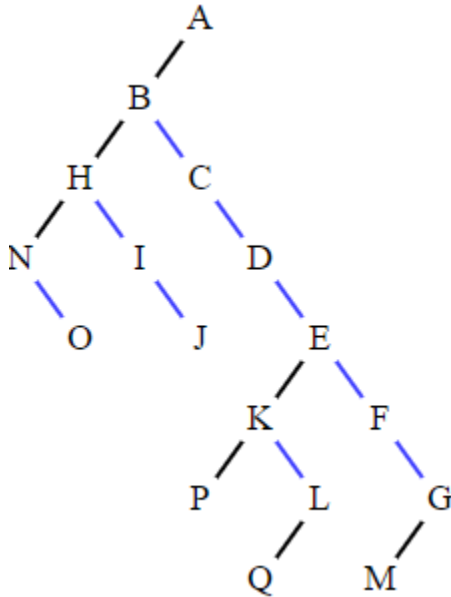
21. Para el siguiente árbol binario:



Presente:

- a. Peso: **11**
- b. Altura: **5**
- c. Hojas: **5**
- d. Una rama: **55**
- e. Recorrido en inorden  
**33-39-48-50-55-57-64-74-76-78-81**
- f. Recorrido en preorden  
**64-55-48-33-39-50-57-76-74-78-81**
- g. Recorrido en postorden  
**39-33-50-48-57-55-74-81-78-76-64**

22. Para el siguiente árbol binario:



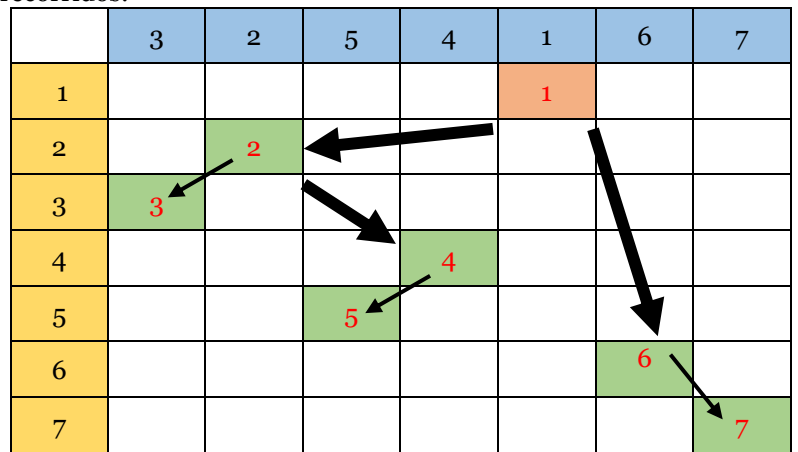
Indique:

- a. Altura: **8**
- b. Número de niveles: **8**
- c. Ancestro común de la E y la J **B**
- d. Peso del árbol izquierdo de la E **4**
- e. Recorrido en inorden
- f. Recorrido en preorden  
**A-B-H-N-O-I-J-C-D-E-K-P-L-Q-F-G-M**
- g. Recorrido en postorden  
**O-N-J-I-H-P-Q-L-K-M-G-F-E-D-C-B-A**
- h. Recorrido por niveles  
**A-B-H-C-N-I-D-O-J-E-K-F-P-L-G-Q-M**
- i. Hojas **5**

23. Reconstruya el árbol binario que posee los siguientes recorridos:

**Preorden:** 1 - 2 - 3 - 4 - 5 - 6 - 7

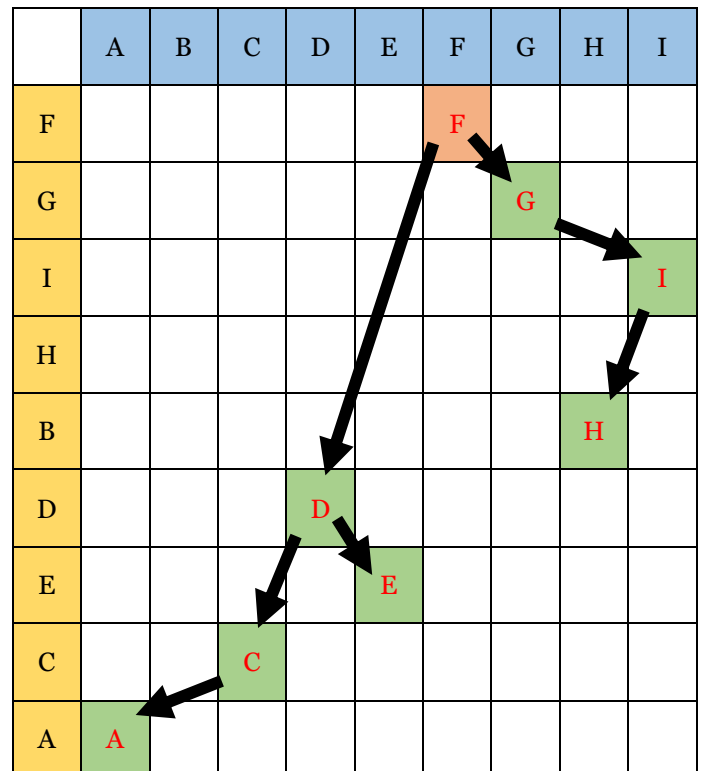
**Inorden:** 3 - 2 - 5 - 4 - 1 - 6 - 7



24. Reconstruya el árbol binario que posee los siguientes recorridos:

**Postorden:** A - C - E - D - B - H - I - G - F

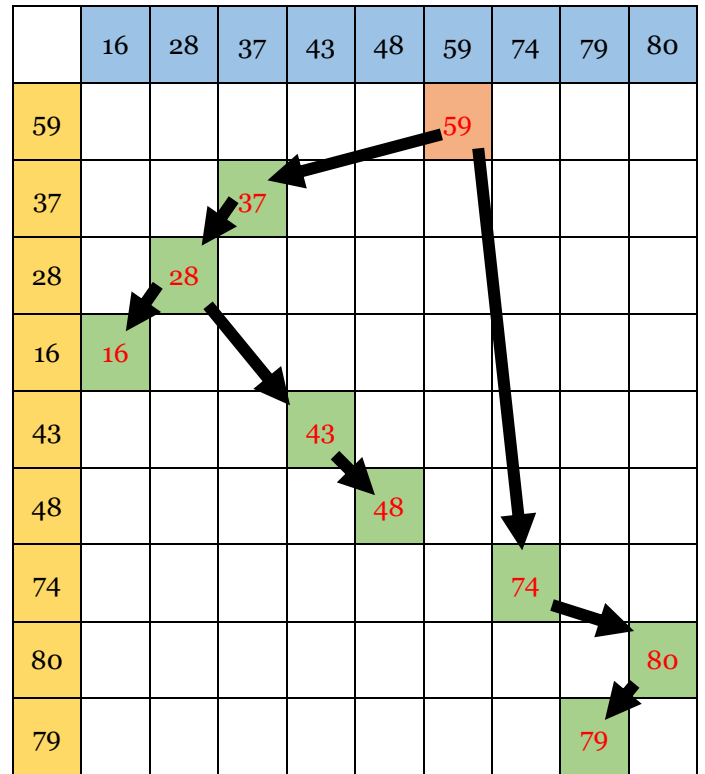
**Inorden:** A - B - C - D - E - F - G - H - I



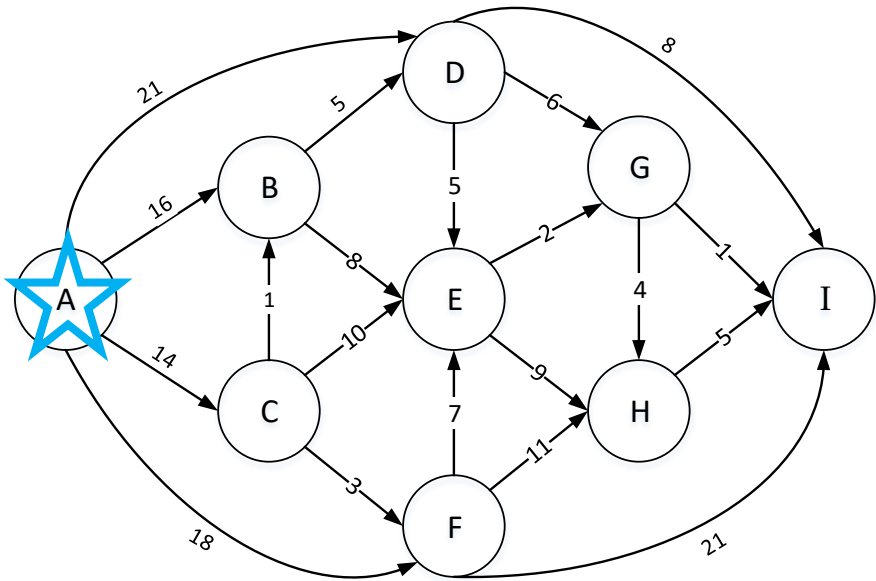
25. Reconstruya el árbol binario que posee los siguientes recorridos:

**Preorden:**  $59 - 37 - 28 - 16 - 43 - 48 - 74 - 80 - 79$

**Inorden:** 16 – 28 – 37 – 43 – 48 – 59 – 74 – 79 – 80



Para el siguiente grafo:



- a. Encuentre (paso a paso) los costos mínimos de los caminos que parten del vértice **A** usando el algoritmo de Dijkstra

Iteración 1:

Vértice	Seleccionados		A	B	C	D	E	F	G	H	I
A	A	Costo	0	16	14	21	∞	18	∞	∞	∞
		Anterior:	-	A	A	A		A			

Iteración 2:

Vértice	Seleccionados		A	B	C	D	E	F	G	H	I
C	A, C	Costo	0	15	14	21	24	17	∞	∞	∞
		Anterior:	-	C	A	A	C	C			

Vecinos de C	Costo anterior	Costo nuevo
B	16	14 + C -> B 15
E	∞	14 + C -> E 24
F	18	14 + C -> F 17

Iteración 3:

Vértice	Seleccionados		A	B	C	D	E	F	G	H	I
B	A, C, B	Costo	0	15	14	20	23	17	∞	∞	∞
		Anterior:	-	C	A	B	B	C			

Vecinos de B	Costo anterior	Costo nuevo
D	21	15 + B -> D 20
E	24	15 + B -> E 23

Iteración 4:

Vértice	Seleccionados		A	B	C	D	E	F	G	H	I
F	A, C, B, F	Costo	0	15	14	20	23	17	∞	28	38
		Anterior:	-	C	A	B	B	C		F	F

Vecinos de F	Costo anterior	Costo nuevo	
E	23	$17 + F \rightarrow E$	24
H	$\infty$	$17 + F \rightarrow H$	28
I	$\infty$	$17 + F \rightarrow I$	38

#### Iteración 5:

Vértice	Seleccionados		A	B	C	D	E	F	G	H	I
D	A, C, B, F, D	Costo	0	15	14	20	23	17	26	28	28
		Anterior:	-	C	A	B	B	C	D	F	D

Vecinos de D	Costo anterior	Costo nuevo	
E	23	$20 + D \rightarrow E$	25
G	$\infty$	$20 + D \rightarrow G$	26
I	38	$20 + D \rightarrow I$	28

#### Iteración 6:

Vértice	Seleccionados		A	B	C	D	E	F	G	H	I
E	A, C, B, F, D, E	Costo	0	15	14	20	23	17	25	28	28
		Anterior:	-	C	A	B	B	C	E	F	D

Vecinos de E	Costo anterior	Costo nuevo	
G	26	$23 + E \rightarrow G$	25
H	28	$23 + E \rightarrow H$	32

#### Iteración 7:

Vértice	Seleccionados		A	B	C	D	E	F	G	H	I
G	A, C, B, F, D, E	Costo	0	15	14	20	23	17	25	28	26
		Anterior:	-	C	A	B	B	C	E	F	G

Vecinos de G	Costo anterior	Costo nuevo	
H	28	$25 + G \rightarrow H$	29
I	28	$25 + G \rightarrow I$	26

#### Iteración 8:

Vértice	Seleccionados		A	B	C	D	E	F	G	H	I
I	A, C, B, F, D, E, I	Costo	0	15	14	20	23	17	25	28	26
		Anterior:	-	C	A	B	B	C	E	F	G

- b. ¿Cuál es el camino entre el vértice A y el vértice I y cuánto es el costo de tal camino, de acuerdo con el algoritmo de Dijkstra?

**El costo mínimo del camino es de 26 pasando por los vertices A, C, B, E, G, I**