

# Algoritmos y Estructuras de Datos

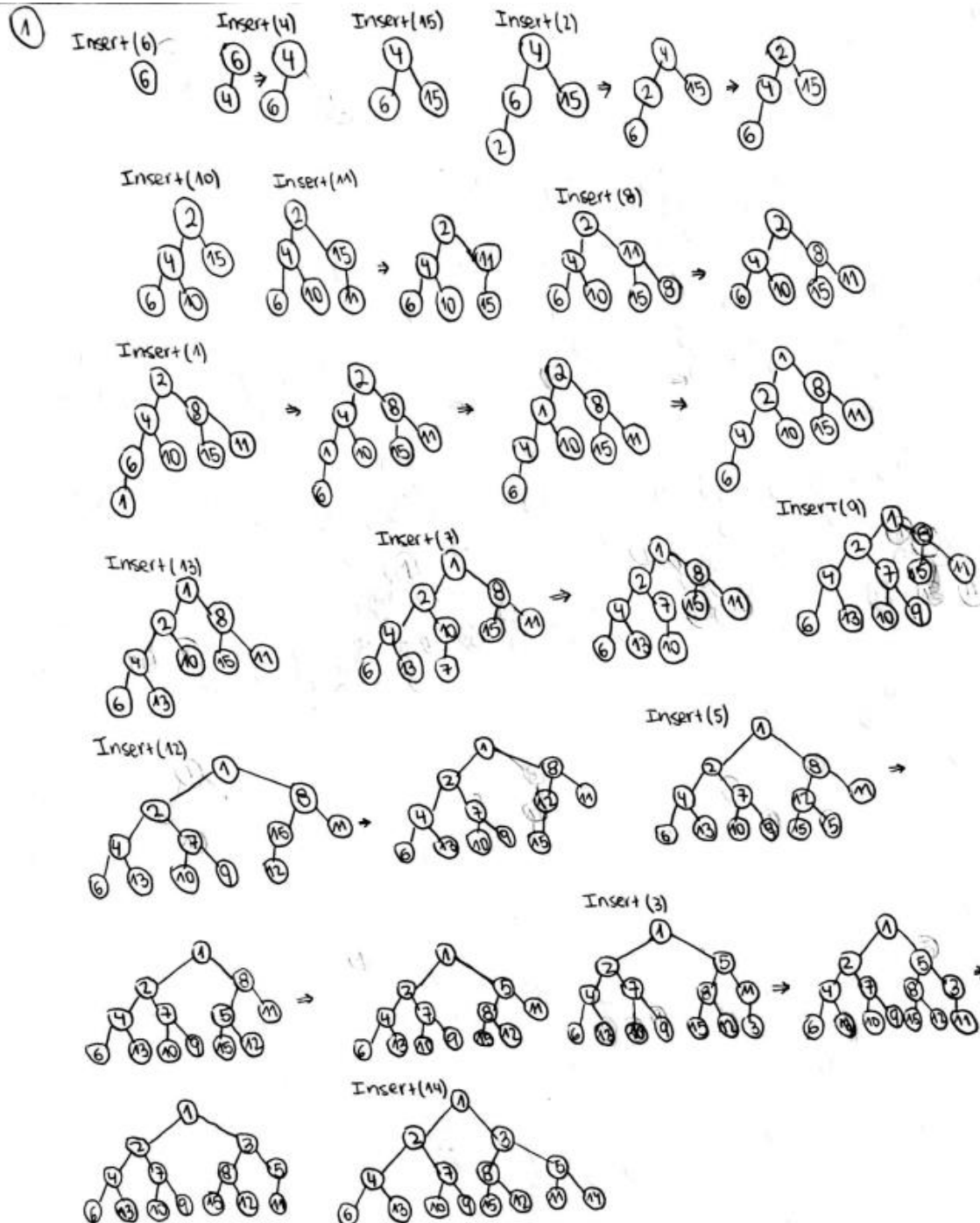
## Cursada 2024

### Ejercitación sobre Colas de Prioridades - Heap

#### Ejercicio 1

A partir de una heap inicialmente vacía, inserte de a uno los siguientes valores:

6, 4, 15, 2, 10, 11, 8, 1, 13, 7, 9, 12, 5, 3, 14



1	2	3	4	7	8	5	6	13	10	9	15	12	11	14
---	---	---	---	---	---	---	---	----	----	---	----	----	----	----

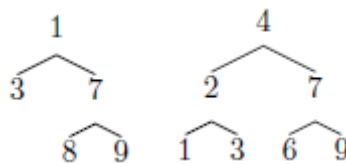
**Ejercicio 2**

- a) ¿Cuántos elementos hay, al menos, en una heap de altura  $h$ ?  
 b) ¿Dónde se encuentra ubicado el elemento mínimo en una max-heap?  
 c) ¿El siguiente arreglo es una max-heap : [23, 17, 14, 6, 13, 10, 1, 5, 7, 12] ?

**Respuestas:** a) En una heap de altura  $h$ , hay al menos  $2^h$  elementos.  
 b) El elemento mínimo en una max-heap se encuentra ubicado en alguna de las hojas.  
 c) El arreglo [23, 17, 14, 6, 13, 10, 1, 5, 7, 12] no es una max-heap debido a que el nodo con valor 7 es padre del nodo con valor 6, no cumpliendo con la propiedad de orden de una max-heap: el dato almacenado en cada nodo es mayor o igual al de sus hijos (6 no es mayor a 7).

**Ejercicio 3**

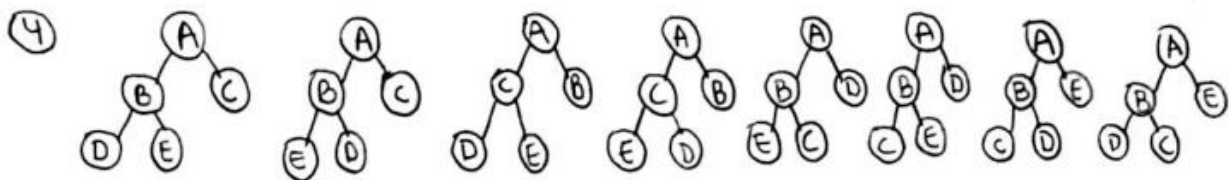
Dados los siguientes árboles, indique si representan una heap. Justifique su respuesta.



**Respuesta:** el primer árbol no representa una heap, ya que no cumple con la propiedad estructural de una heap: Un árbol binario completo de altura  $h$  es un árbol binario lleno de altura  $h-1$  y en el nivel  $h$ , los nodos se completan de izquierda a derecha. Como indica la propiedad, los nodos se completan de izquierda a derecha, y en este caso, en el último nivel se estarían completando en orden inverso.

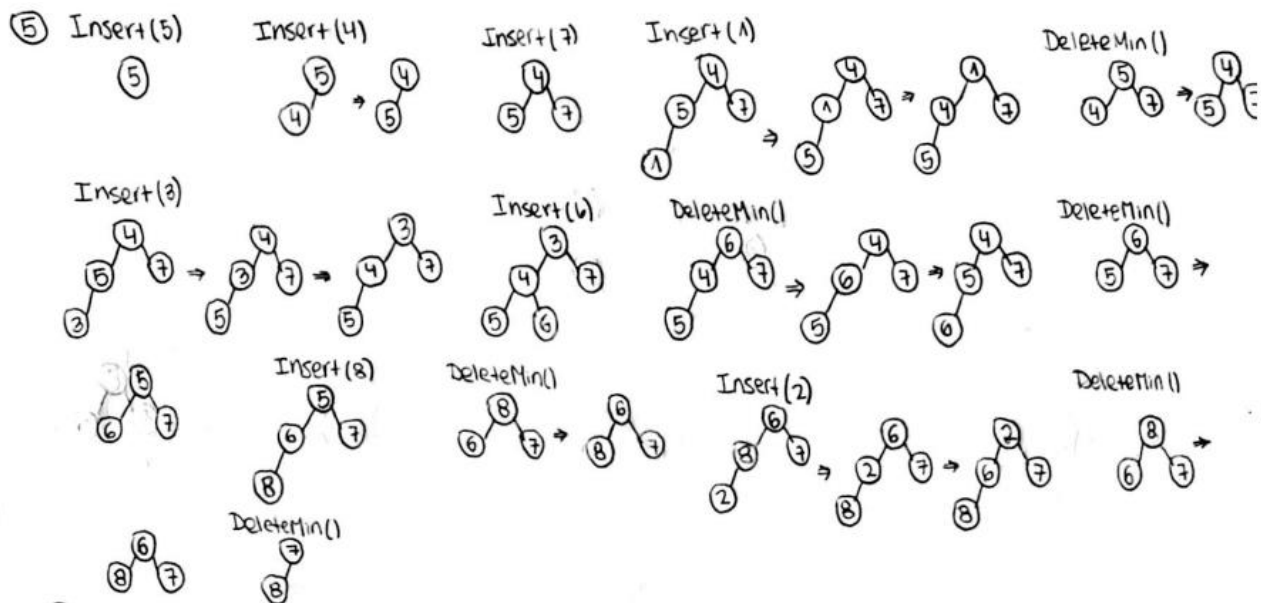
**Ejercicio 4**

Dibuje todas las min-heaps posibles para este conjunto de claves: {A, B, C, D, E}

**Ejercicio 5**

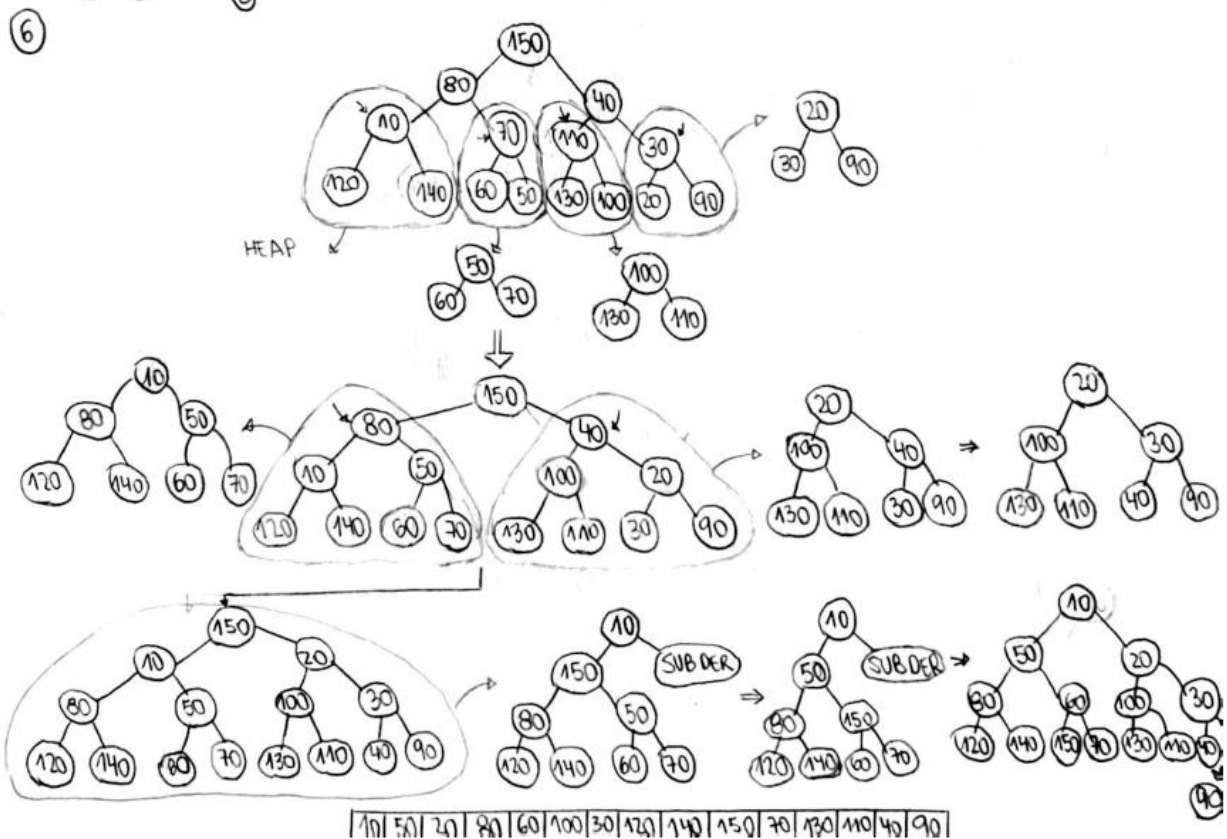
A partir de una min-heap inicialmente vacía, dibuje la evolución del estado de la heap al ejecutar las siguientes operaciones:

Insert(5), Insert(4), Insert(7), Insert(1), DeleteMin(), Insert(3), Insert(6), DeleteMin(), DeleteMin(), Insert(8), DeleteMin(), Insert(2), DeleteMin(), DeleteMin()



### Ejercicio 6

Aplique el algoritmo *BuildHeap*, para construir una min-heap en **tiempo lineal**, con los siguientes valores {150, 80, 40, 10, 70, 110, 30, 120, 140, 60, 50, 130, 100, 20, 90}



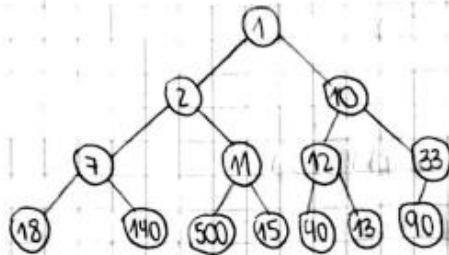
### Ejercicio 7

Aplique el algoritmo *HeapSort*, para ordenar descendentemente los siguientes elementos:

{15, 18, 40, 1, 7, 10, 33, 2, 140, 500, 11, 12, 13, 90}

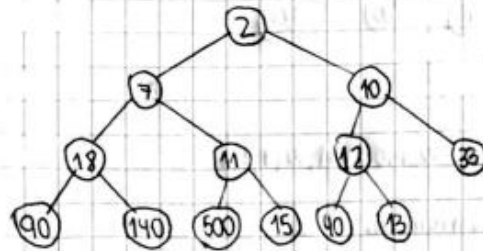
Muestre paso a paso la ejecución del algoritmo sobre los datos.

⑦



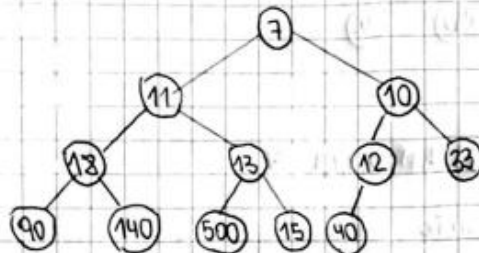
El arreglo actual es  $\{1, 2, 10, 7, 11, 12, 33, 18, 140, 500, 15, 40, 13, 90\}$

Intercambio el primero con último y decremento el tamaño.



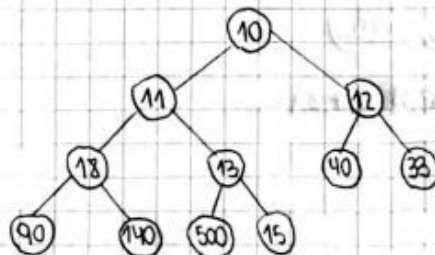
El arreglo actual es  $\{2, 7, 10, 18, 11, 12, 33, 90, 140, 500, 15, 40, 13\} | 1$

Intercambio el primero con el último y decremento el tamaño.



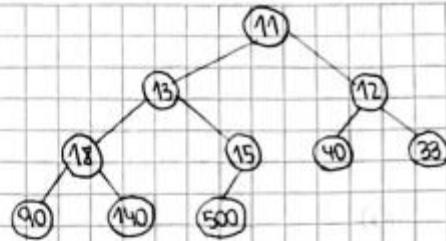
El arreglo actual es:  $\{7, 11, 10, 18, 13, 12, 33, 90, 140, 500, 15, 40\} | 2, 1$

Intercambio el primero con el último y decremento el tamaño.



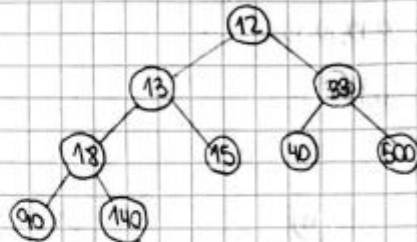
El arreglo actual es:  $\{10, 11, 12, 18, 13, 40, 33, 90, 140, 500, 15\} | 7, 2, 1$

Intercambio el primero con el último y decremento el tamaño.



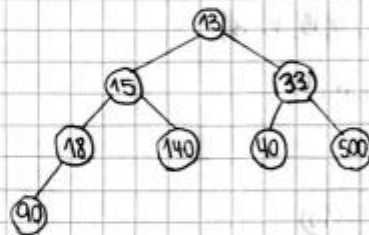
El arreglo actual es:  $\{11, 13, 12, 18, 15, 40, 33, 90, 140, 500\} | 10, 9, 8, 7, 6, 5, 4, 3, 2, 1$

Intercambio el primero con el último y decremento el tamaño.



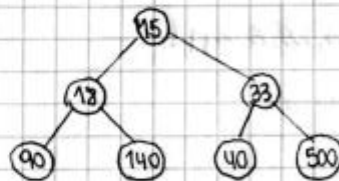
El arreglo actual es:  $\{12, 13, 33, 18, 15, 40, 500, 90, 140\} | 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1$

Intercambio el primero con el último y decremento el tamaño.



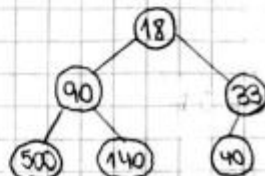
El arreglo actual es:  $\{13, 15, 33, 18, 140, 40, 500, 90\} | 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1$

Intercambio el primero con el último y decremento el tamaño.



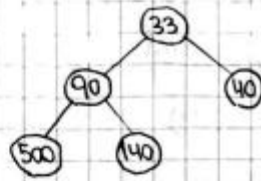
El arreglo actual es:  $\{15, 18, 33, 90, 140, 40, 500\} | 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1$

Intercambio el primero con el último y decremento el tamaño.



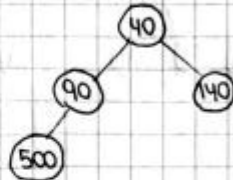
El arreglo actual es:  $\{18, 90, 33, 500, 140, 40\} | 15, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1$

Intercambio el primero con el último y decremento el tamaño.



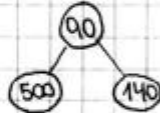
El arreglo actual es:  $\{33, 90, 40, 500, 140\} | 18, 15, 13, 12, 11, 10, 7, 2, 1$

Intercambio el primero con el último y decremento el tamaño.



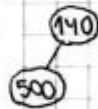
El arreglo actual es:  $\{40, 90, 140, 500\} | 33, 18, 15, 13, 12, 11, 10, 7, 2, 1$

Intercambio el primero con el último y decremento el tamaño.



El arreglo actual es:  $\{90, 500, 140\} | 40, 33, 18, 15, 13, 12, 11, 10, 7, 2, 1$

Intercambio el primero con el último y decremento el tamaño.



El arreglo actual es:  $\{140, 500\} | 90, 40, 33, 18, 15, 13, 12, 11, 10, 7, 2, 1$

Intercambio el primero con el último y decremento el tamaño.



El arreglo actual es:  $\{500\} | 140, 90, 40, 33, 18, 15, 13, 12, 11, 10, 7, 2, 1$

El arreglo actual tiene un tamaño de uno y se agrega al arreglo ordenado.

Resultado:  $\{500, 140, 90, 40, 33, 18, 15, 13, 12, 11, 10, 7, 2, 1\}$

### Ejercicio 8

Construir una max-heap binaria con los siguientes datos:

$\{5, 8, 12, 9, 7, 10, 21, 6, 14, 4\}$

a) Insertándolos de a uno

b) Usando el algoritmo BuildHeap

8) a) Insert(5)

5

Insert(8)

8  
5

Insert(12)

12  
5 8

Insert(9)

12  
9 8  
5

Insert(7)

12  
9 8  
5 7

Insert(10)

12  
9 10  
5 7 8

Insert(21)

21  
9 12  
5 7 8 10

Insert(6)

21  
9 12  
6 7 8 10  
5

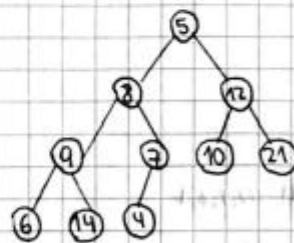
Insert(14)

21  
14 12  
9 7 8 10  
5 6

Insert(4)

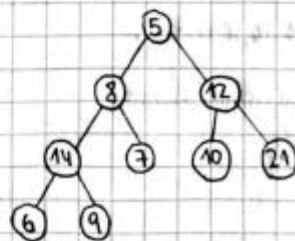
21  
14 12  
9 7 8 10  
5 6 4

b) Se crea un árbol binario con los elementos del arreglo.



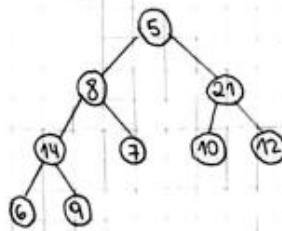
Se empieza a filtrar desde tamaño 12 (7). El 7 está en la posición correcta, por ende se avanza a la anterior posición 12

arreglo (9) y se filtra.

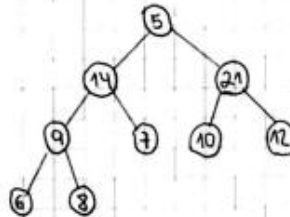




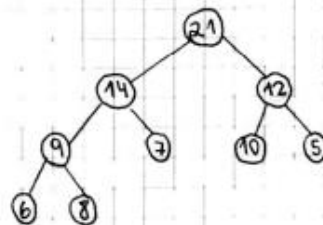
Se filtra la posición anterior del arreglo, el 12.



Se filtra la posición anterior del arreglo, el 8.



Se filtra la posición anterior del arreglo, el 5.



### Ejercicio 9

Suponga que una heap que representa una cola de prioridades está almacenada en el arreglo A (se comienza de la posición A[1]). Si insertamos la clave 16, ¿en qué posición quedará?

i:	1	2	3	4	5	6	7	8	9	10	11	12
A[i]:	11	21	27	37	36	34	32	43	44	42	51	62

- (a) A[2]      (b) A[3]      (c) A[6]      (d) A[7]      (e) A[12]

### Ejercicio 10

Suponga que una heap que representa una cola de prioridades está almacenada en el arreglo A (se comienza de la posición A[1]). Si aplica un delete-min, ¿en qué posición quedará la clave 62?

i:	1	2	3	4	5	6	7	8	9	10	11	12
A[i]:	11	21	27	37	36	34	32	43	44	42	51	62

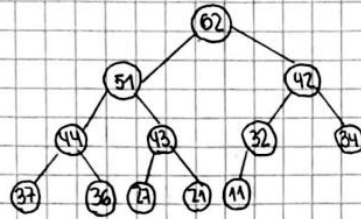
- (a) A[1]      (b) A[2]      (c) A[10]      (d) A[11]      (e) A[12]

### Ejercicio 11

- Ordenar en forma creciente los datos del ejercicio anterior, usando el algoritmo HeapSort.
- ¿Cuáles serían los pasos a seguir si se quiere ordenar en forma decreciente?

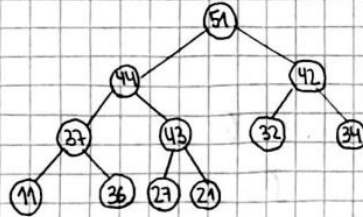


(11) a)



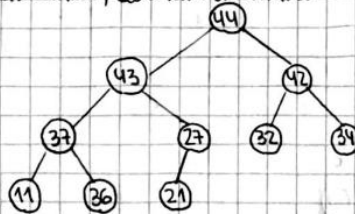
El arreglo actual es:  $\{62, 51, 42, 44, 43, 32, 34, 37, 36, 27, 21, 11\}$

Intercambio el primero con el último y decremento el tamaño.



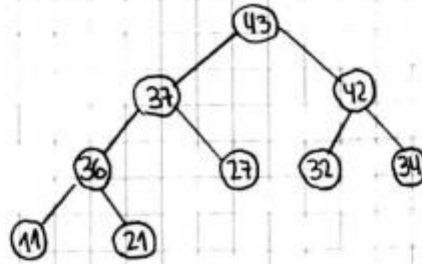
El arreglo actual es:  $\{51, 44, 42, 37, 43, 32, 34, 11, 36, 27\}$  | 62

Intercambio el primero con el último y decremento el tamaño.

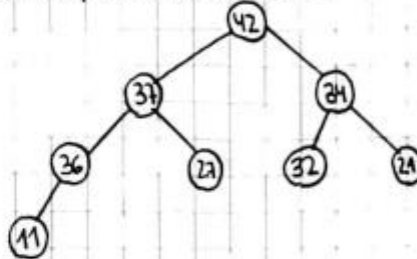


El arreglo actual es:  $\{44, 43, 42, 37, 27, 32, 34, 11, 36, 21\}$  | 51, 62

Intercambio el primero con el último y decremento el tamaño.

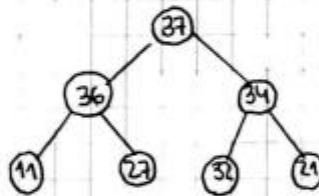


El arreglo actual es:  $\{43, 37, 42, 36, 27, 32, 34, 11, 21\} | 44, 51, 62$   
 Intercambio el primero con el último y decremento el tamaño.



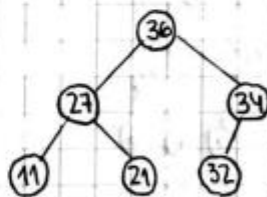
El arreglo actual es:  $\{42, 37, 34, 36, 27, 32, 21, 11\} | 43, 44, 51, 62$

Intercambio el primero con el último y decremento el tamaño.

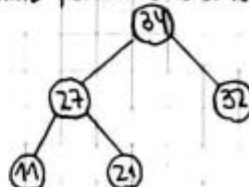


El arreglo actual es:  $\{37, 36, 34, 11, 27, 32, 21\} | 42, 43, 44, 51, 62$

Intercambio el primero con el último y decremento el tamaño.



El arreglo actual es:  $\{36, 27, 34, 11, 21, 32\} | 37, 42, 43, 44, 51, 62$   
 Intercambio el primero con el último y decremento el tamaño.



El arreglo actual es:  $\{34, 27, 32, 11, 21\} | 36, 37, 42, 43, 44, 51, 62$

Intercambio el primero con el último y decremento el tamaño.

**Ejercicio 12**

¿Cuáles de los siguientes arreglos representan una max-heap, min-heap o ninguna de las dos?

- arreglo 1: 0 1 2 0 4 5 6 7 8 9
- arreglo 2: 9 8 7 6 5 4 3 2 1 0
- arreglo 3: 5 5 5 6 6 6 6 7 7 1
- arreglo 4: 9 3 9 2 1 6 7 1 2 1
- arreglo 5: 8 7 6 1 2 3 4 2 1 2

**Respuesta:** Solamente los arreglos 2 y 4 son max-heap, y los demás ninguna de las dos.

**Ejercicio 13**

Un arreglo de 7 enteros se ordena ascendentemente usando el algoritmo *HeapSort*. Luego de la fase inicial del algoritmo (la construcción de la heap), ¿cuál de los siguientes es un posible orden del arreglo?

- (a) 85 78 45 51 53 47 49
- (b) 85 49 78 45 47 51 53**
- (c) 85 78 49 45 47 51 53
- (d) 45 85 78 53 51 49 47
- (e) 85 51 78 53 49 47 45

**Ejercicio 14**

En una Heap, ¿para un elemento que está en la posición  $i$  su hijo derecho está en la posición.....?

- (a)  $\lfloor i/2 \rfloor$
- (b)  $2*i$
- (c)  $2*i + 1$**
- (d) Ninguna de las anteriores

**Ejercicio 15**

¿Siempre se puede decir que un árbol binario lleno es una Heap?

- (a) Sí
- (b) No**

**Ejercicio 16**

La operación que agrega un elemento a la heap que tiene  $n$  elementos, en el peor caso es de .....

- (a)  $O(n)$
- (b)  $O(n \log n)$
- (c)  $O(\log n)$**
- (d) Ninguna de las otras opciones

**Ejercicio 17**

Se construyó una Máx-Heap con las siguientes claves: 13, 21, 87, 30, 25, 22, 18. ¿Cuál de las siguientes opciones corresponde al resultado de realizar la construcción insertando las claves **una a una**?

- (a) 87, 30, 25, 22, 21, 18, 13
- (b) 87, 30, 22, 21, 25, 13, 18
- (c) 87, 30, 25, 13, 22, 18, 21
- (d) 87, 30, 22, 13, 25, 21, 18**

**Ejercicio 18**

Se construyó una Máx-Heap con las siguientes claves: 13, 21, 87, 30, 25, 22, 18. ¿Cuál de las siguientes opciones corresponde al resultado de realizar la construcción aplicando el algoritmo **Build-Heap**?

- (a) 87, 30, 25, 22, 21, 18, 13
- (b) 87, 30, 22, 21, 25, 13, 18**
- (c) 87, 30, 25, 13, 22, 18, 21
- (d) 87, 30, 22, 13, 25, 21, 18

**Ejercicio 19**

El algoritmo HeapSort consta de dos etapas:

- 1) se construye una heap y
- 2) se realizan los intercambios necesarios para dejar ordenados los datos.

Asuma que la heap ya está construida y es la siguiente:

58 38 53 23 28 40 35 18

¿Cómo quedan los datos en el arreglo después de ejecutar sólo 2 pasos de la segunda etapa del Heapsort?

- (a) 40 38 23 28 35 18 53 58
- (b) 53 38 40 23 28 18 35 58
- (c) 40 38 23 35 28 18 53 58
- (d) 40 38 35 23 28 18 53 58

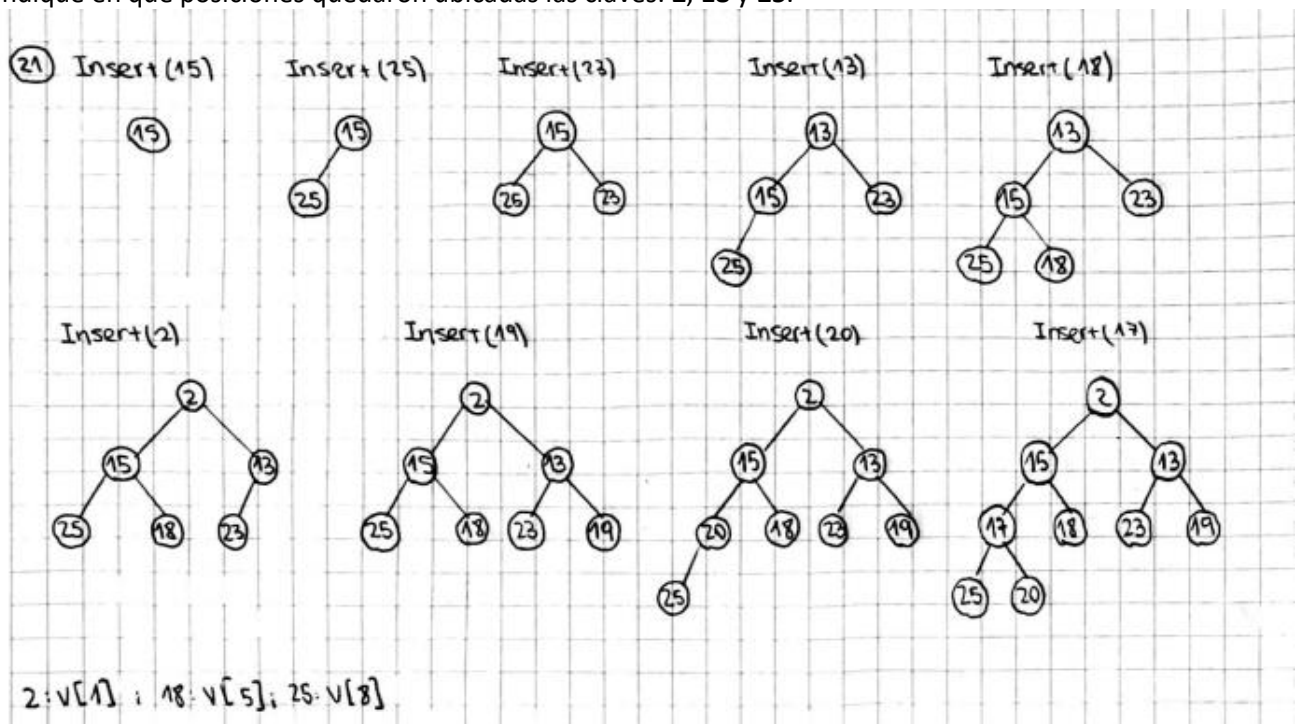
**Ejercicio 20**

Dada la Min-Heap 3, 8, 5, 15, 10, 7, 19, 28, 16, 25, 12. ¿En qué **posición** está ubicado el hijo derecho de la clave 15?

- (a) 7
- (b) 8
- (c) 9
- (d) 10

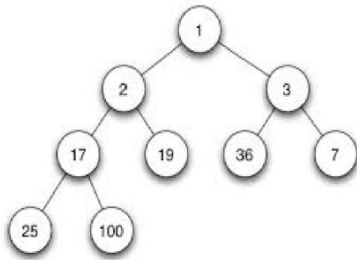
**Ejercicio 21**

Construya una min-heap con las siguientes claves: 15, 25, 23, 13, 18, 2, 19, 20, 17 insertándose una a una. Indique en qué posiciones quedaron ubicadas las claves: **2, 18 y 25**.



**Ejercicio 22**

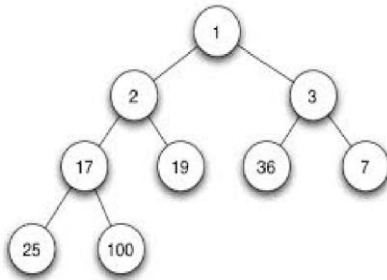
Luego de insertar la clave **15** en la siguiente min-heap, ¿cuántas de las claves que ya estaban en la heap han mantenido su lugar (es decir, ocupan en la min-heap resultante la misma posición que ocupaban antes de la inserción)?



- a) Ninguna
- b) Seis
- c) Ocho**
- d) Nueve

**Ejercicio 23**

Luego de una operación de borrado del mínimo en la siguiente min-heap, ¿cuántas claves han cambiado de lugar (es decir, ocupan en la min-heap resultante un lugar diferente al que ocupaban en la min-heap antes del borrado) ? (No contar la clave borrada, ya que no pertenece más a la heap)



- a) Ninguno
- b) Dos
- c) Tres
- d) Cuatro**