

Heapsort

MARKS CALDERÓN NIQUIN
UNIVERSIDAD

Heapsort

Heapsort llamado montículo. La idea central de este algoritmo se basa en dos operaciones:

1. Construir un montículo
2. Eliminar la raíz del montículo en forma repetida

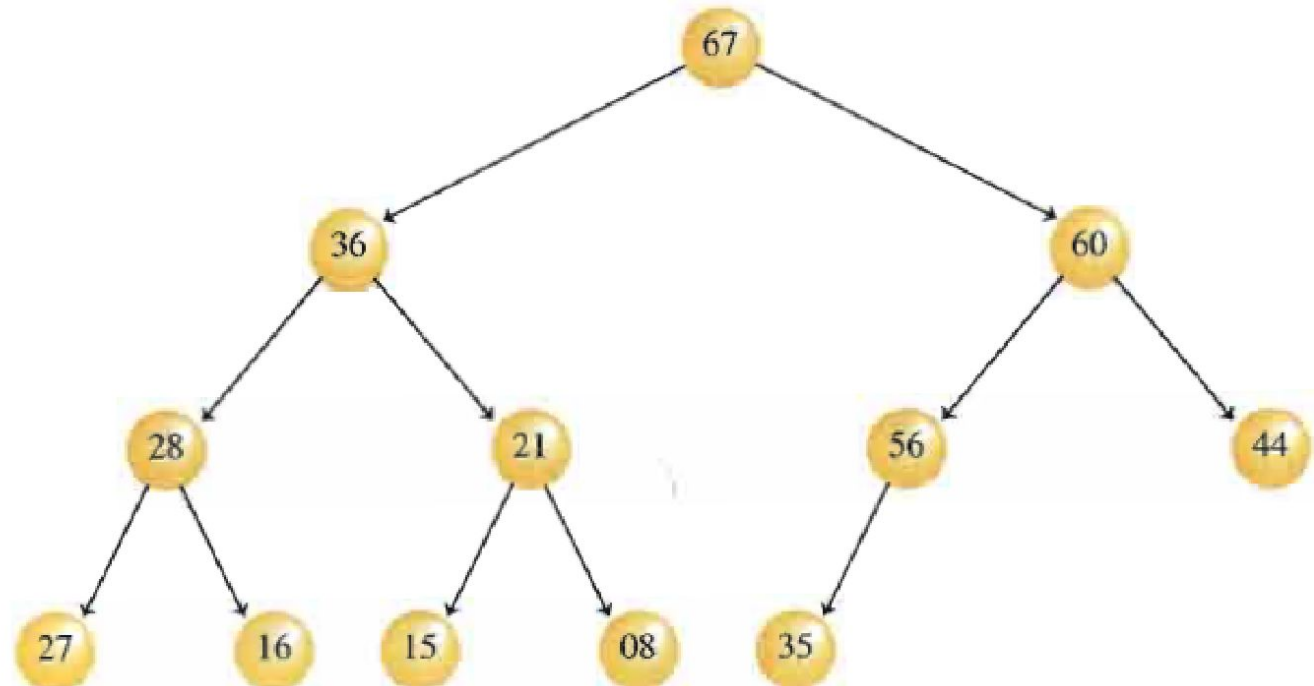
Para todo nodo del árbol se debe cumplir que su valor sea mayor o igual que el valor de cualquiera de sus hijos.

Heapsort

Un montículo en un arreglo lineal se debe tener en cuenta para todo nodo K lo siguiente:

- El nodo K se almacena en la posición K correspondiente del arreglo.
- El hijo izquierdo del nodo K se almacena en la posición $2*K$
- El hijo derecho del nodo K se almacena en la posición $2*K+1$

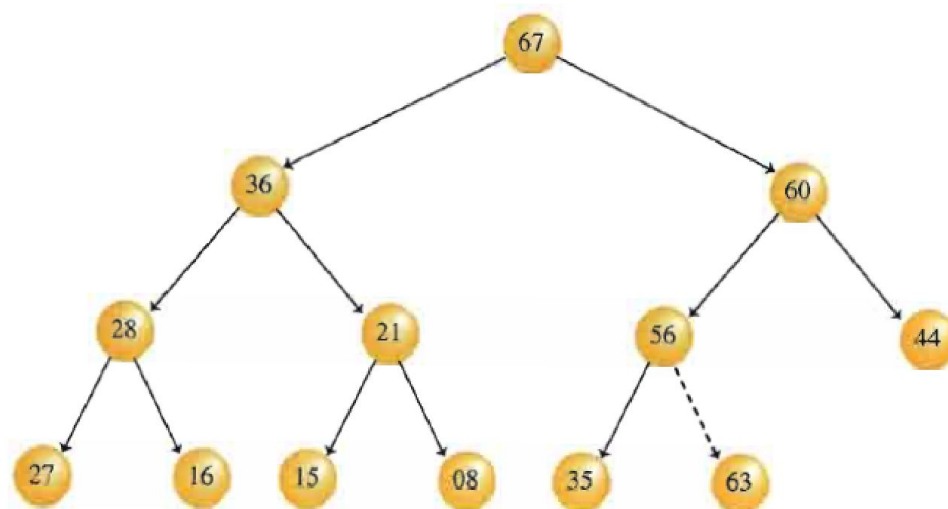
A	67	36	60	28	21	56	44	27	16	15	08	35
	1	2	3	4	5	6	7	8	9	10	11	12



Inserción de un elemento en un montículo

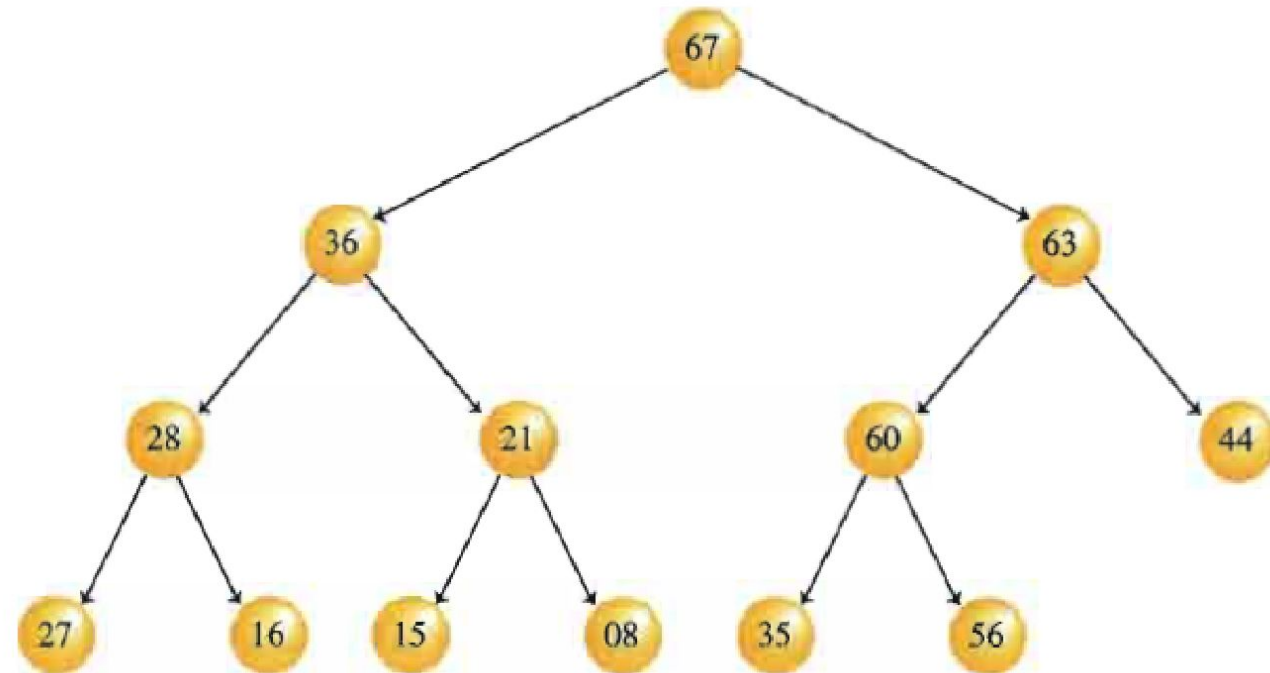
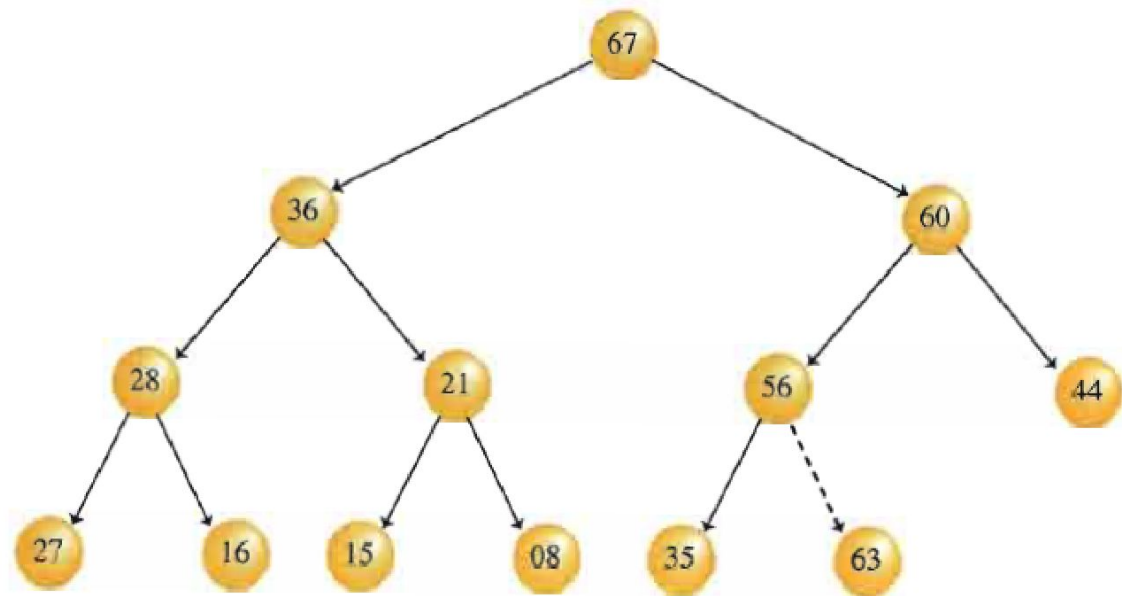
La inserción de un elemento en un montículo se lleva a cabo por medio de los siguientes pasos:

1. Se inserta el elemento en la primera posición disponible.
2. Se verifica si su valor es mayor que el de su padre. Si se cumple esta condición, entonces se efectúa el intercambio. Si no se cumple esta condición, entonces el algoritmo se detiene y el elemento queda ubicado en su posición correcta al montículo



63 > 56 sí hay intercambio
63 > 60 sí hay intercambio
63 > 67 no hay intercambio

Inserción de un elemento en un montículo



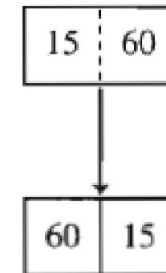
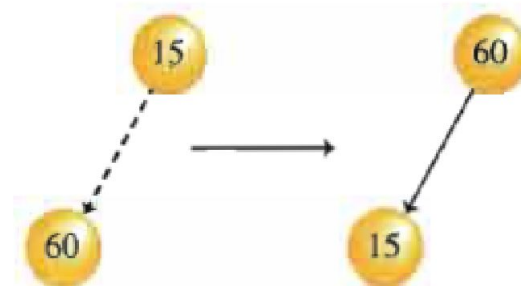
Ejemplo de inserción

Insertamos las siguientes claves:

a) INSERCIÓN: CLAVE 15



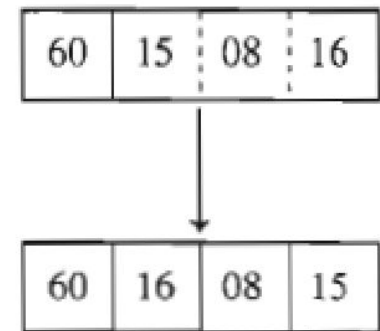
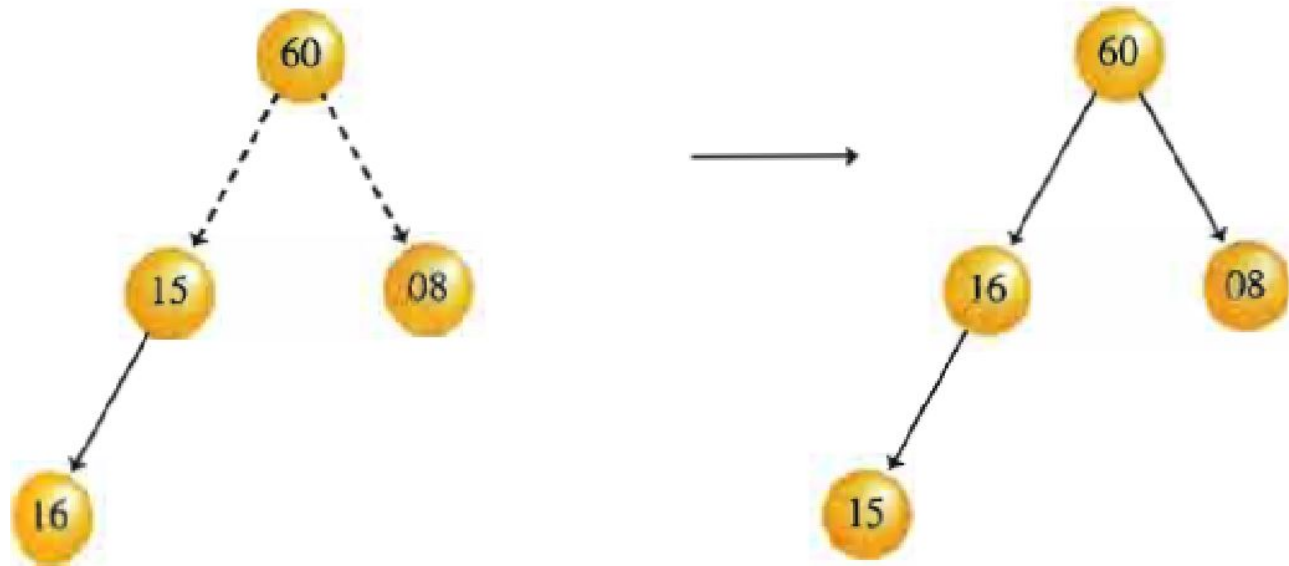
b) INSERCIÓN: CLAVE 60



Ejemplo de inserción

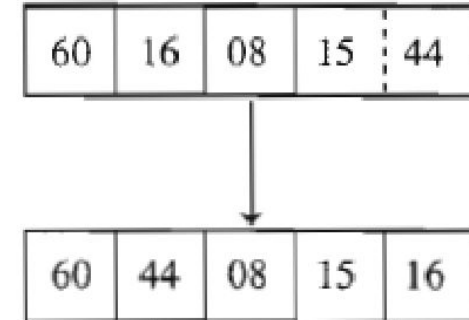
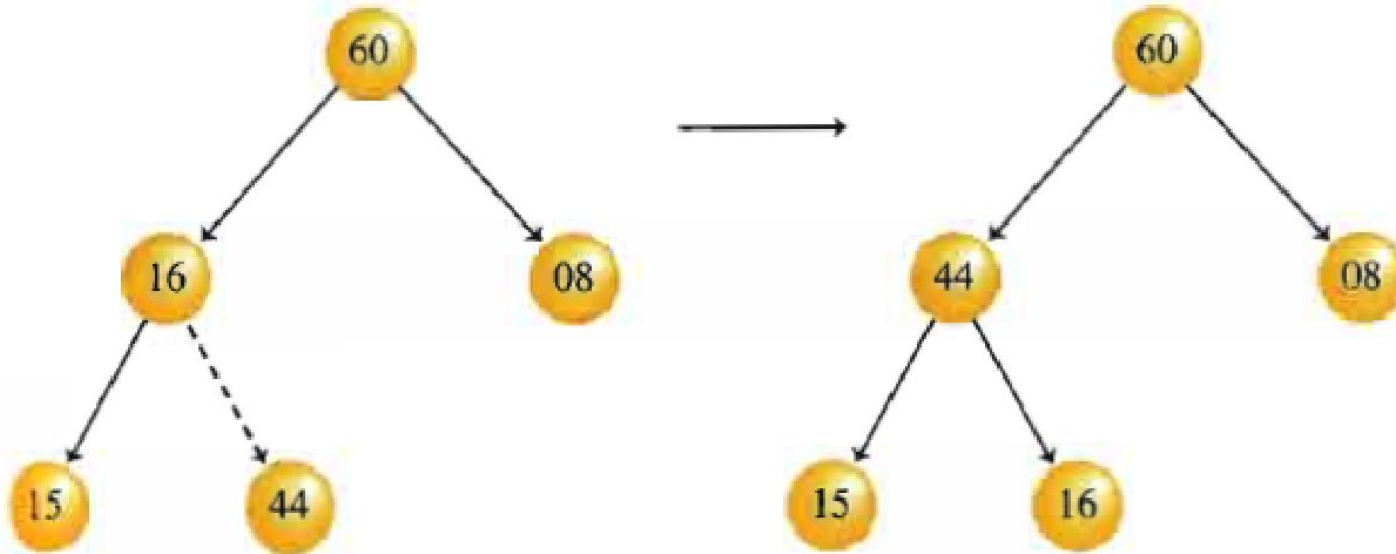
Insertamos las siguientes claves:

⇨ INSERCIÓN: CLAVES 08 y 16



Ejemplo de inserción

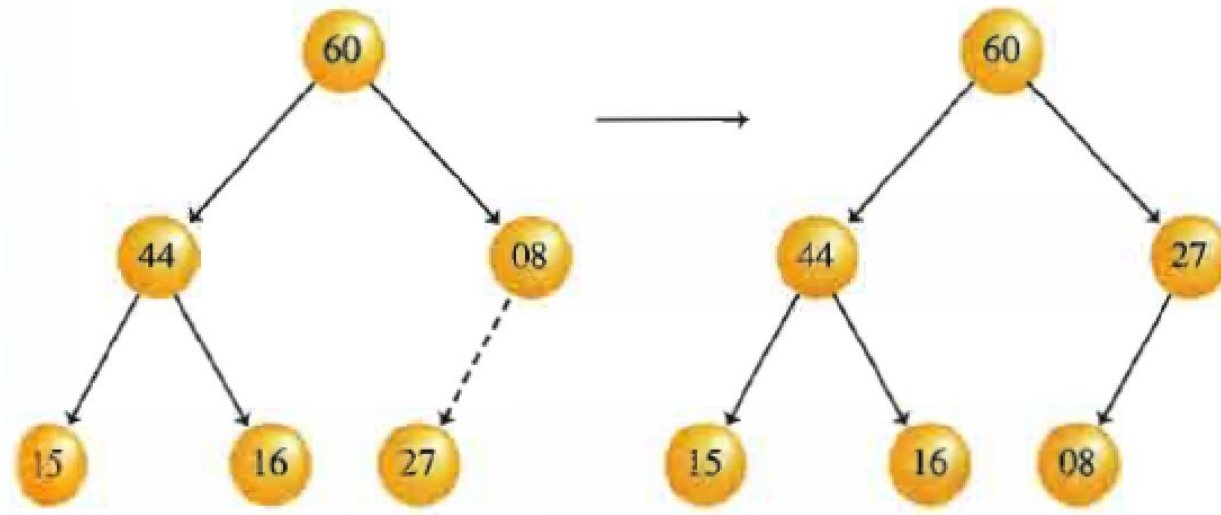
Insertamos las siguientes claves:



Ejemplo de inserción

Insertamos las siguientes claves:

2) INSERCIÓN: CLAVE 27

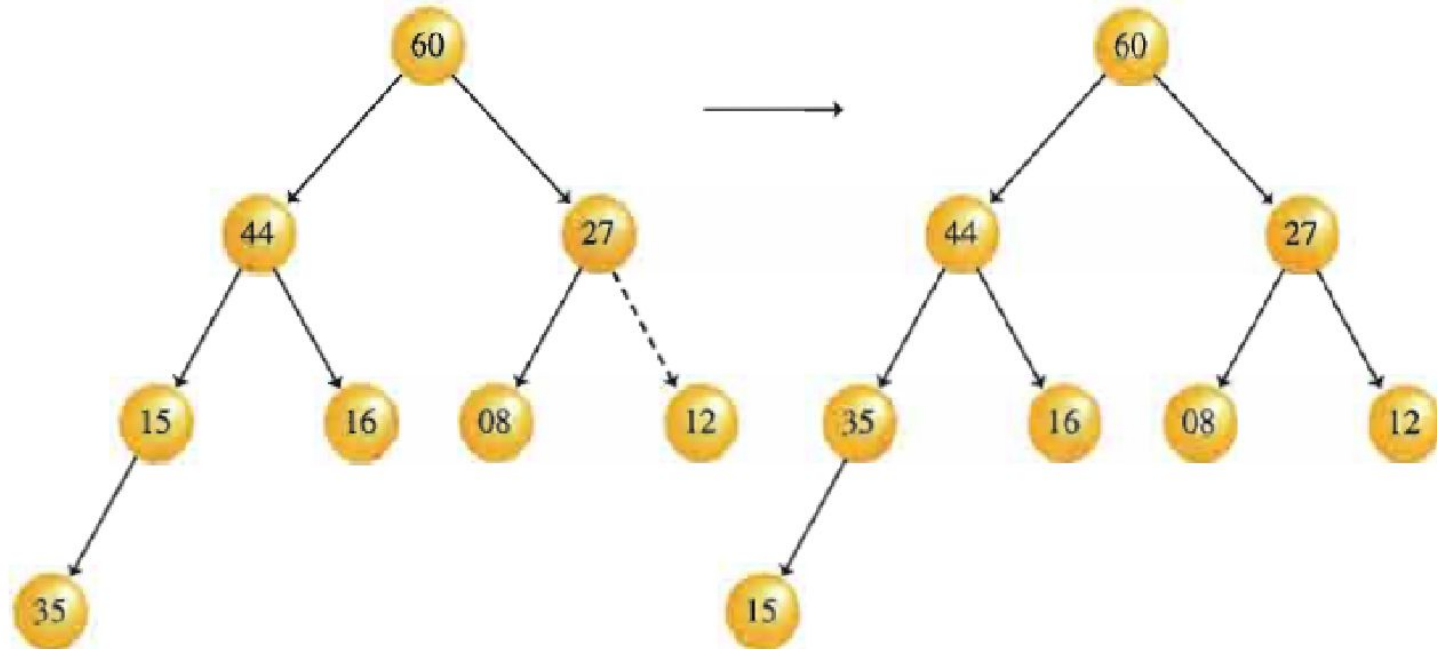


60	44	08	15	16	27
60	44	27	15	16	08

Ejemplo de inserción

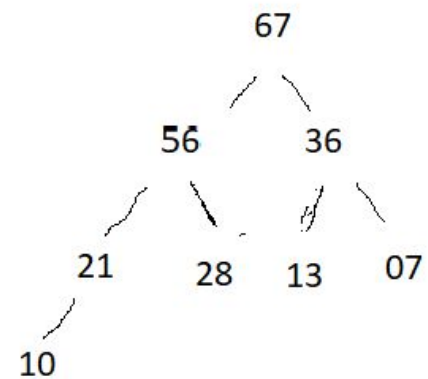
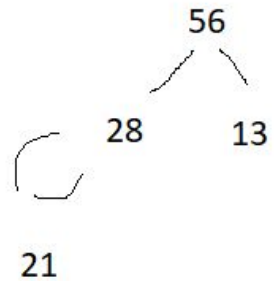
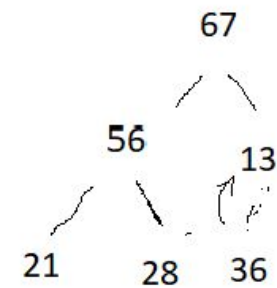
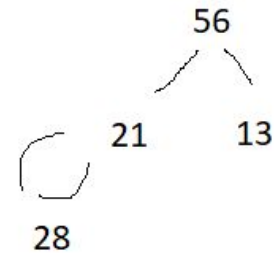
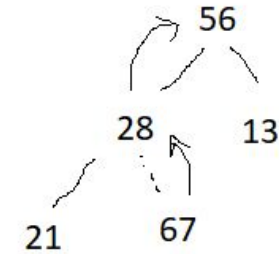
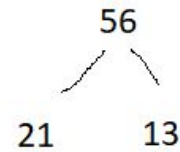
Insertamos las siguientes claves:

f) INSERCIÓN: CLAVES 12 y 35



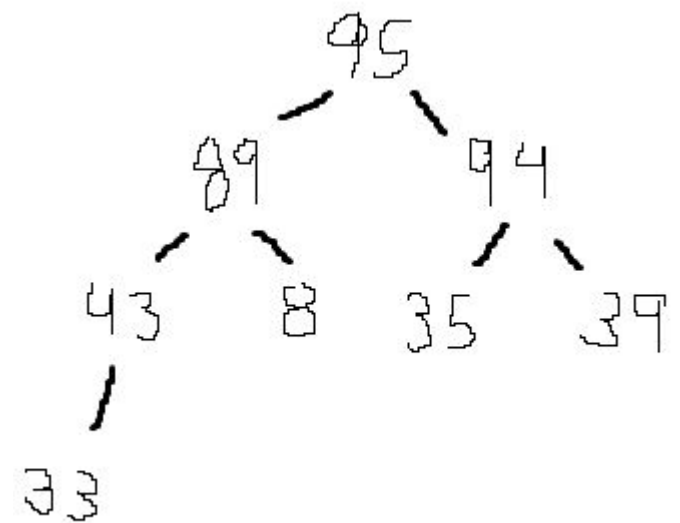
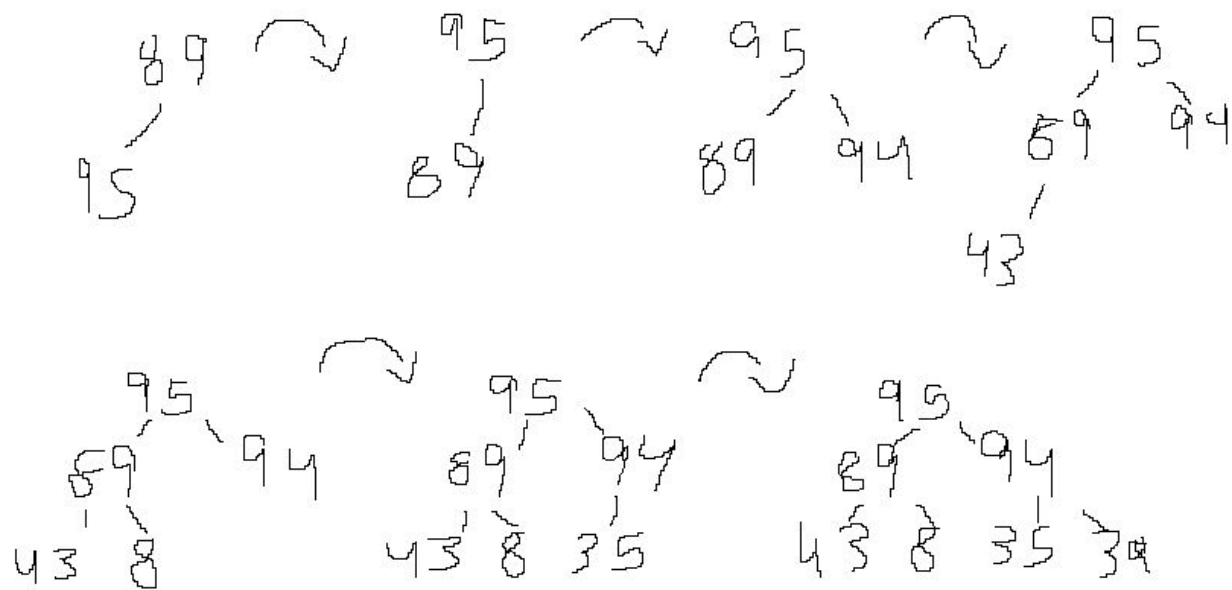
Insertar [Edson]

56 21 13 28 67 36 07 10



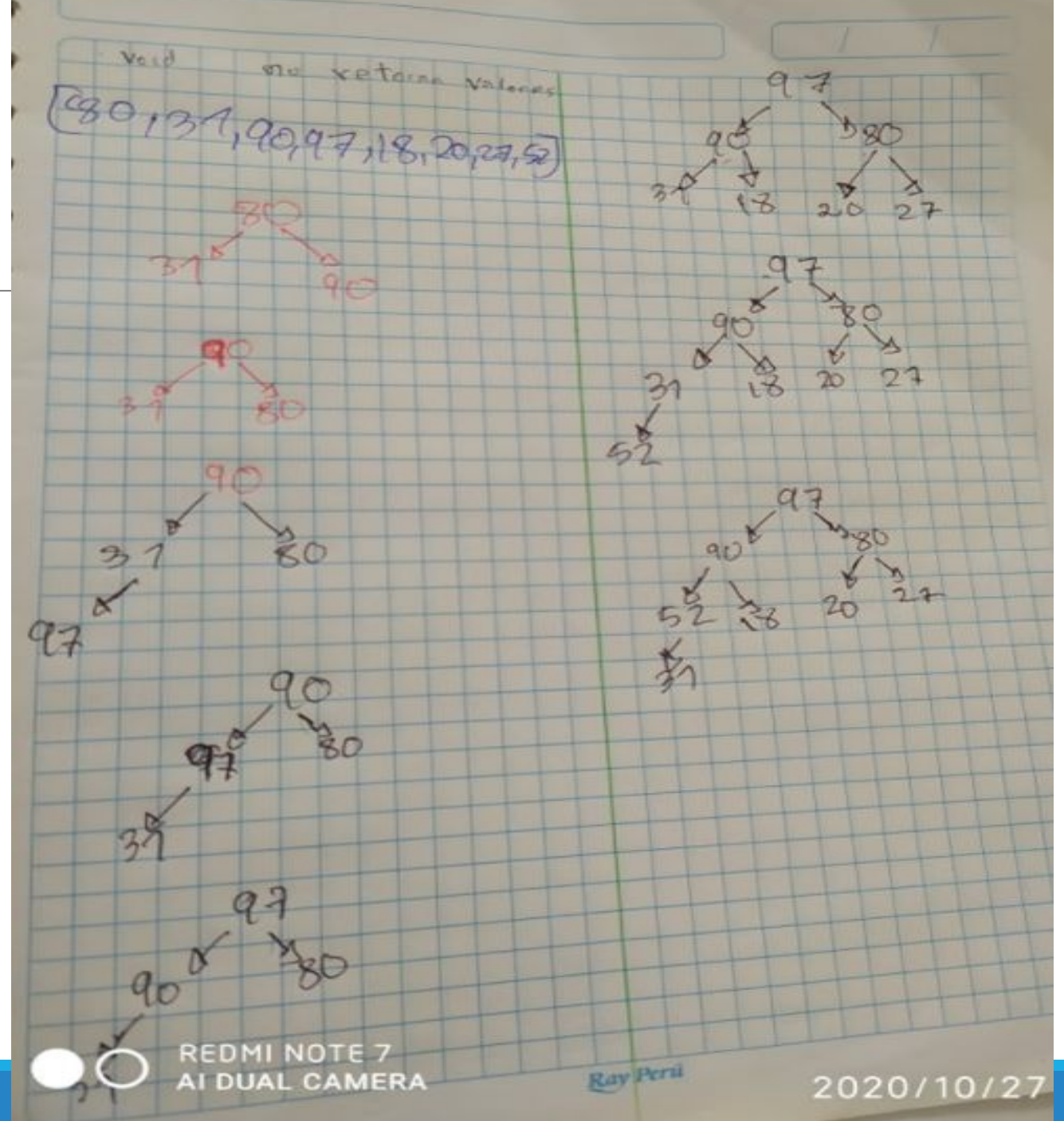
Insertar [Gustavo]

89 95 94 43 8 35 39 33



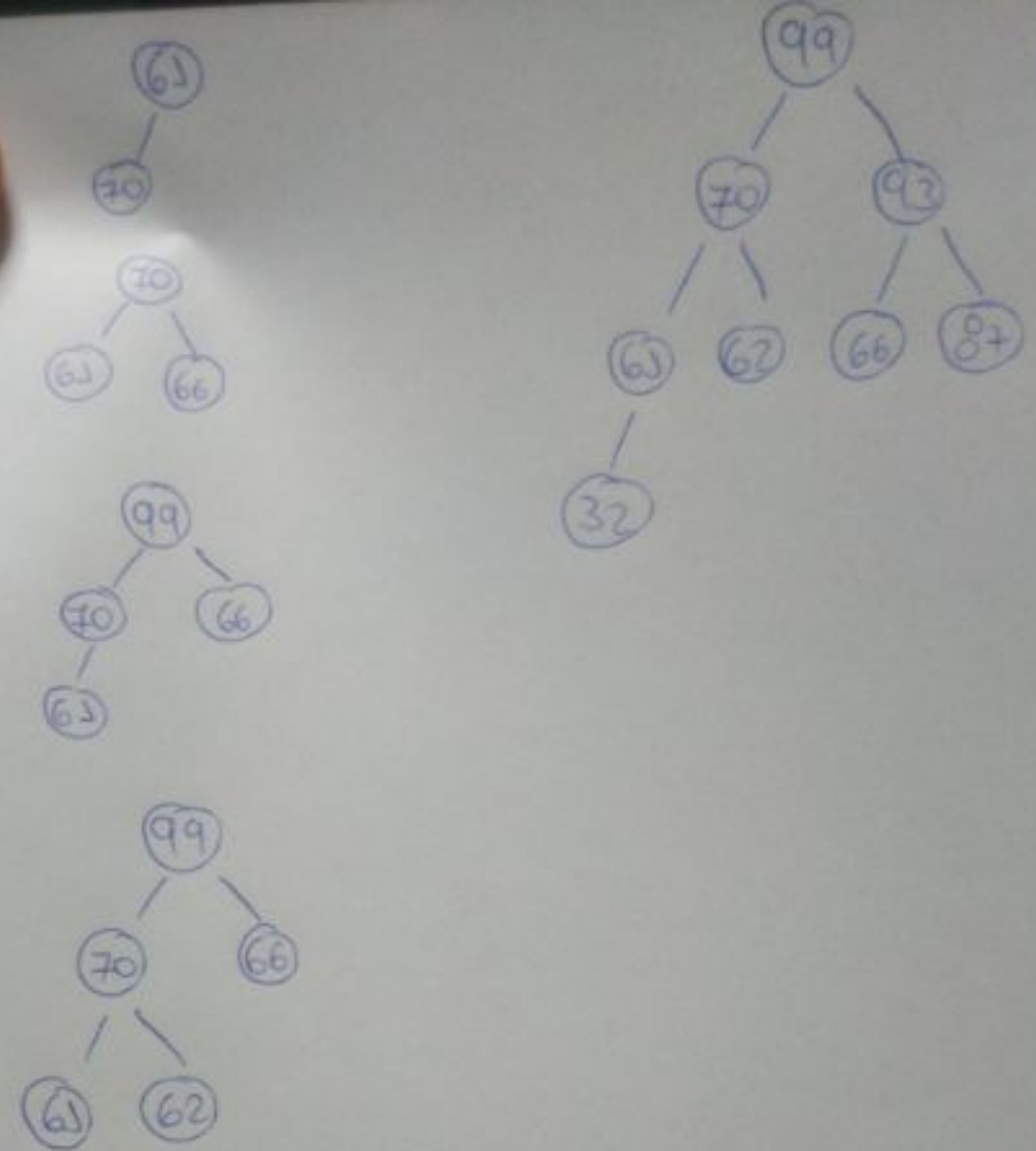
Insertar [Johao]

80 31 90 97 18 20 27 52



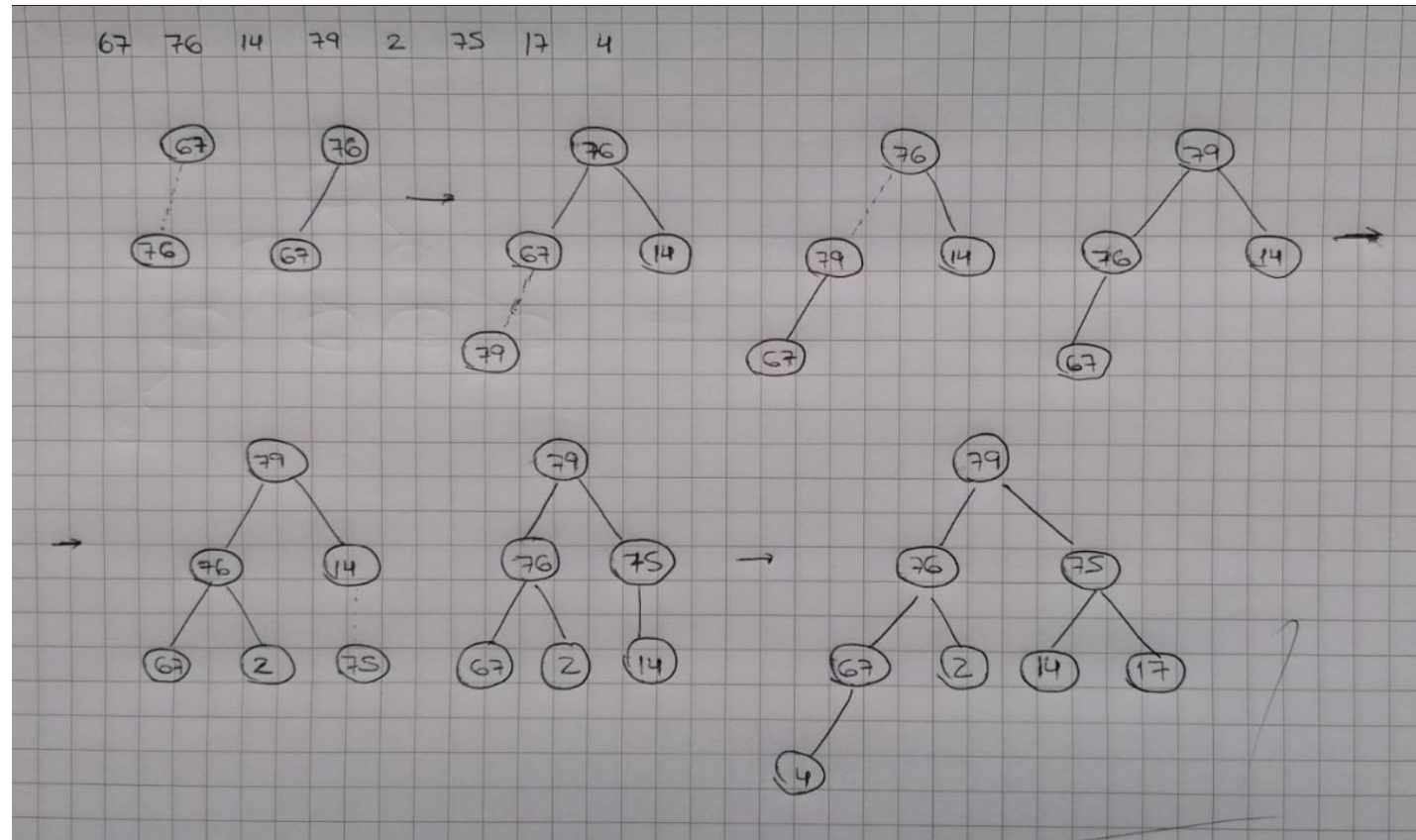
Insertar [Leonardo]

61 70 66 99 62 93 87 32



Insertar [Luis]

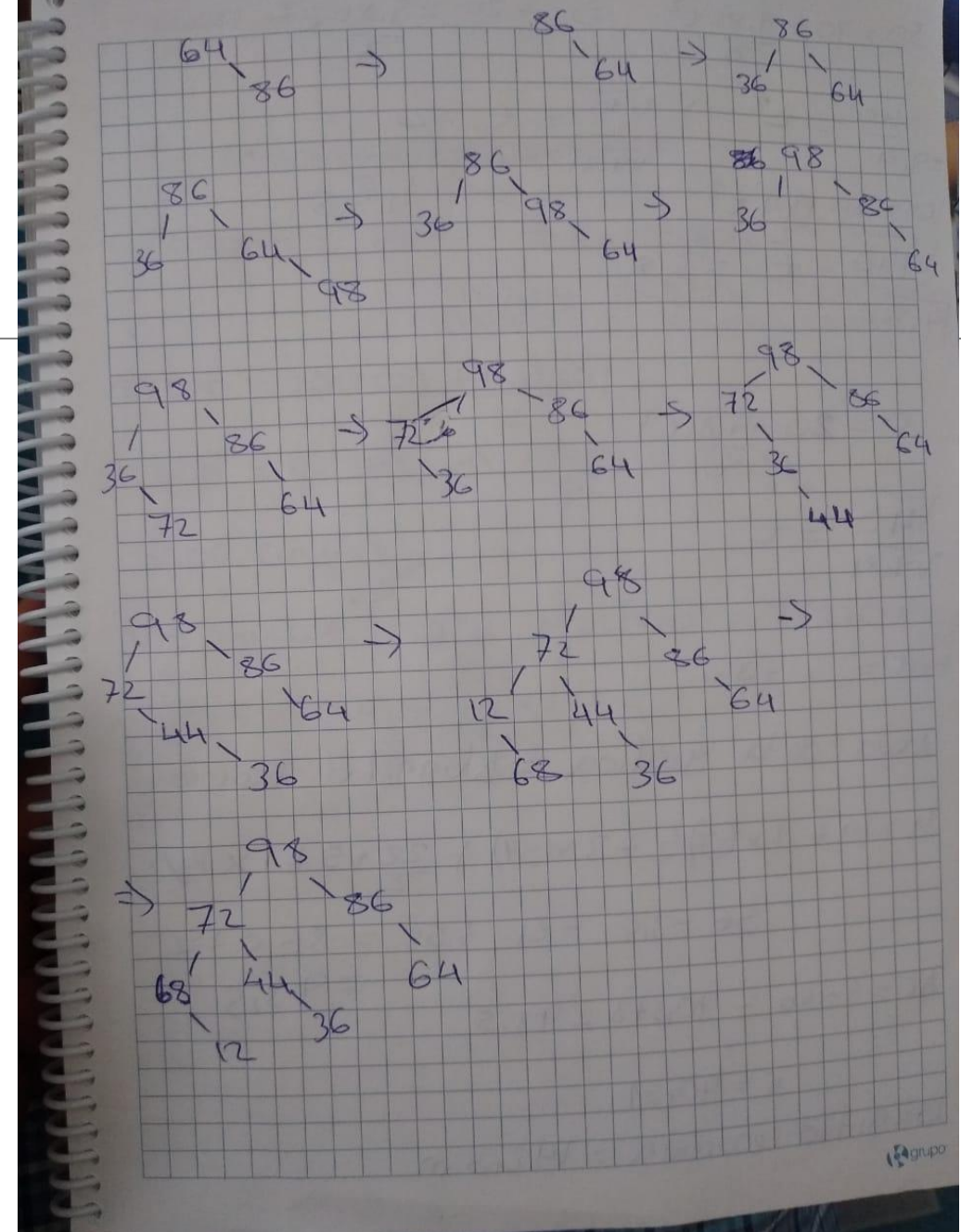
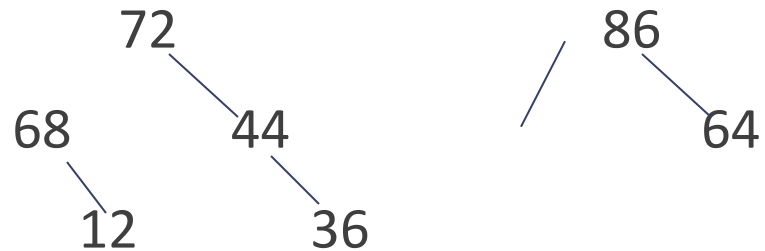
67 76 14 79 2 75 17 4



Insertar [Martin]

64 86 36 98 72 44 12 68

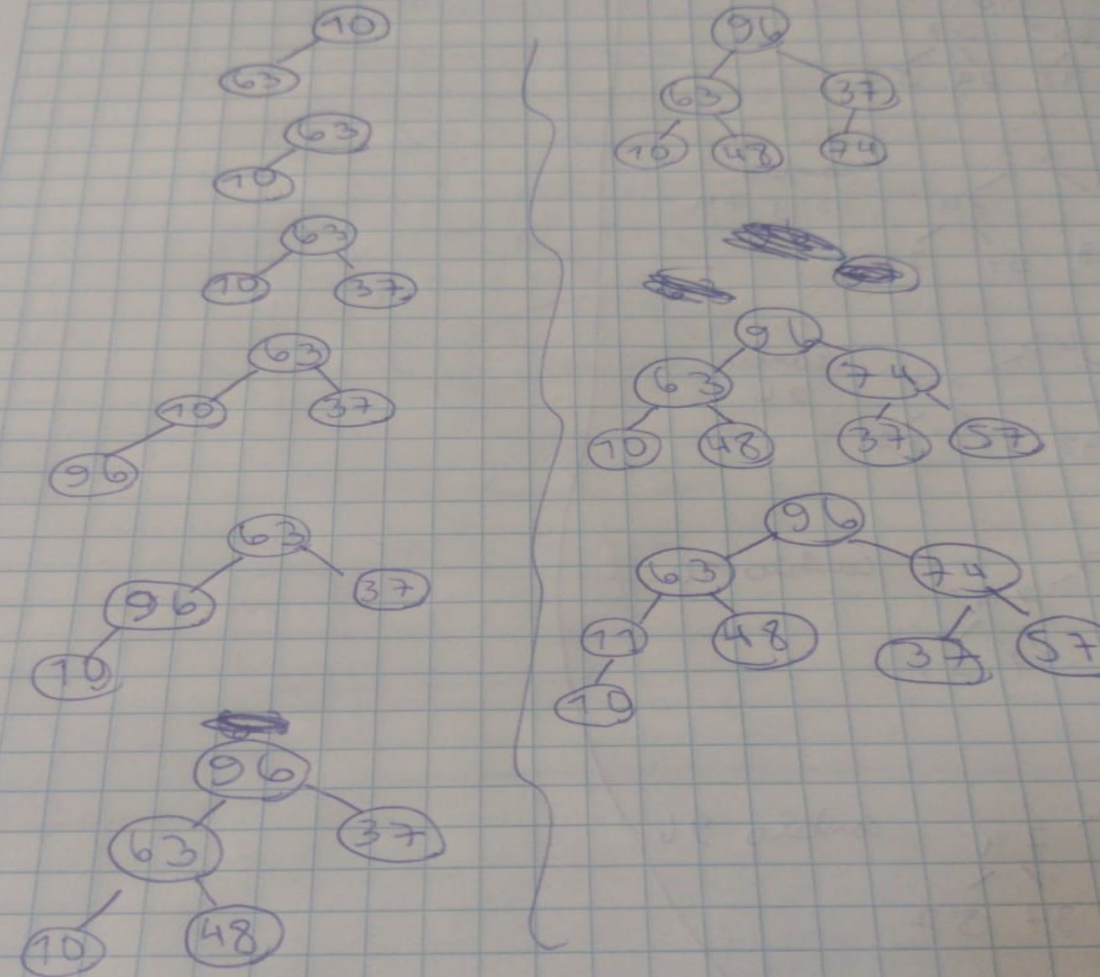
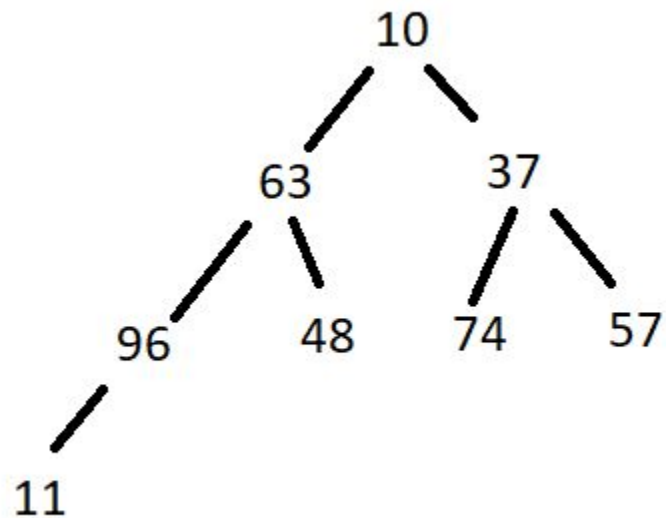
98



Insertar [Ricardo]

10 63 37 96 48 74 57 11

Antes



Insertar montículo

Insertar_monticulo(A,N)

Enteros i,k, aux

Booleano band

Para i=1 hasta N

 k= i band = verdadero

 Mientras ((k>1) y (band==verdadero))

 band = falso

 Si (A[k] > A[piso(k/2)]) entonces

 aux = A[piso(k/2)]

 A[piso(k/2)] = A[k]

 A[k] = aux

 k = piso(k/2)

 band = verdadero

 Fin_si

Fin_mientras

Fin_para

Eliminación de un montículo

Pasos para eliminar la raíz del montículo los pasos son:

1. Se reemplaza la raíz con el elemento que ocupa la última posición del montículo.
2. Se verifica si el valor de la raíz es menor que el valor más grande de sus hijos. Si se cumple la condición, entonces se efectúa el intercambio. Si no se cumple la condición, entonces el algoritmo se detiene y el elemento queda ubicado en su posición correcta en el montículo.

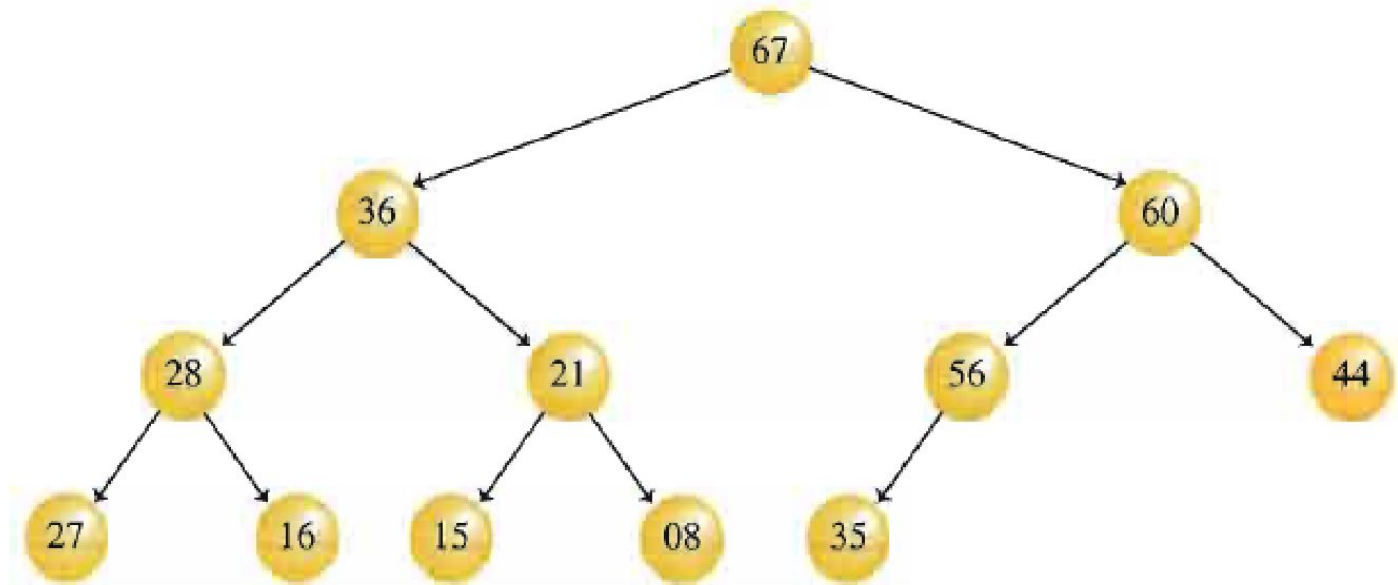
El paso 2 se aplica de manera recursiva y desde arriba hacia abajo.

Ejemplo: eliminación

Se desea eliminar la raíz del montículo (67). A continuación efectuamos las siguientes comparaciones:

35 < 60 si hay intercambio

35 < 56 si hay intercambio

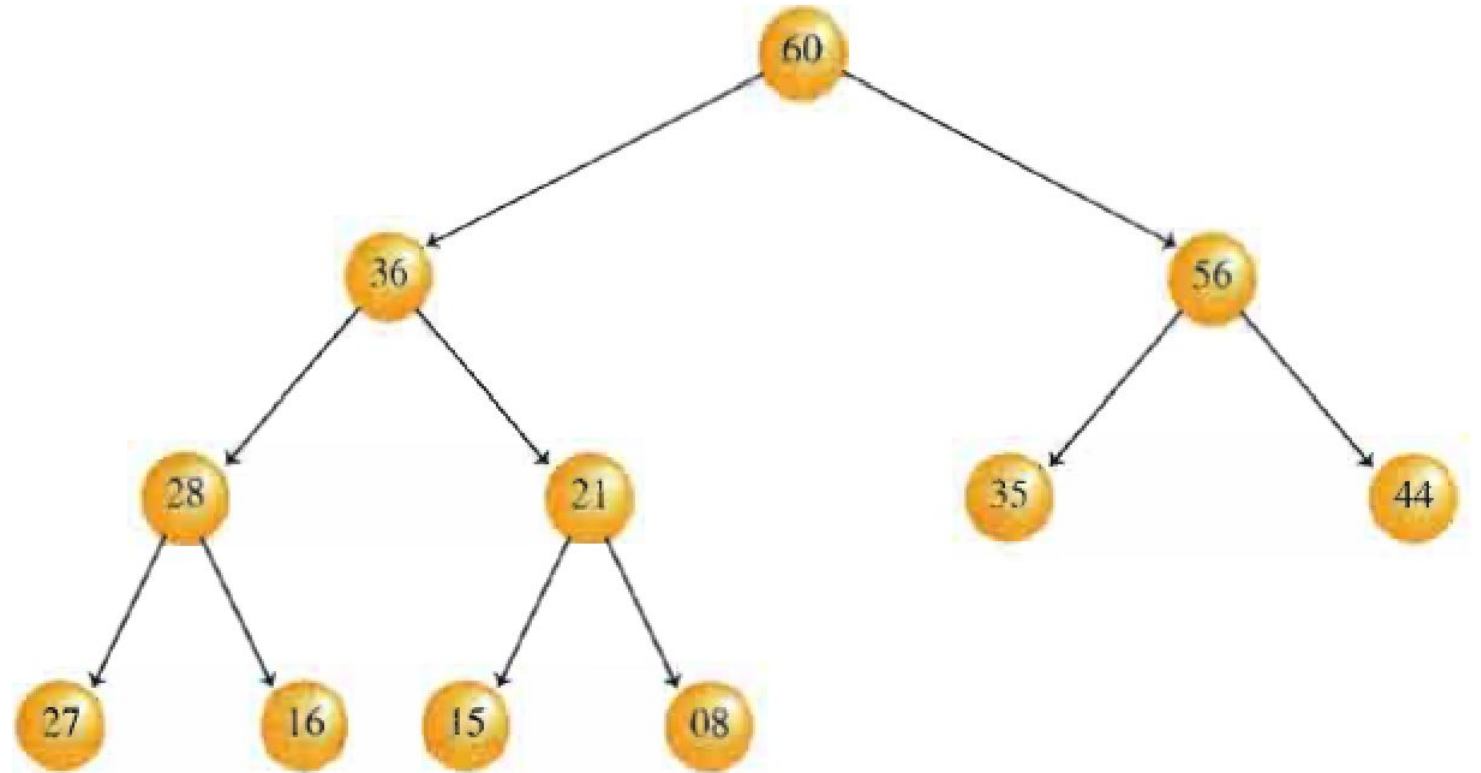


Ejemplo: eliminación

Se desea eliminar la raíz del montículo (67). A continuación efectuamos las siguientes comparaciones:

$35 < 60$ si hay intercambio

$35 < 56$ si hay intercambio



Primera eliminación

PRIMERA ELIMINACIÓN DE LA RAÍZ

Se intercambia la raíz, 67, con el elemento que ocupa la última posición del montículo.
10. Las comparaciones que se realizan son:

$A[1] < A[3]$ $(10 < 60)$ sí hay intercambio
 $A[3]$ es el mayor de los hijos de $A[1]$

$A[3] < A[7]$ $(10 < 36)$ sí hay intercambio
 $A[7]$ es el mayor de los hijos de $A[3]$

$A[7] < A[14]$ $(10 < 12)$ sí hay intercambio
 $A[14]$ es el mayor de los hijos de $A[7]$

Luego de eliminar la primera raíz, el montículo queda así:

60 56 36 44 21 28 12 15 35 16 13 08 27 10 07 67

Segunda eliminación

SEGUNDA ELIMINACION DE LA RAIZ

Se intercambia la raíz, 60, con el elemento que ocupa la última posición del montículo.
07. Las comparaciones que se realizan son:

$A[1] < A[2]$ $(07 < 56)$ sí hay intercambio

$A[2]$ es el mayor de los hijos de $A[1]$

$A[2] < A[4]$ $(07 < 44)$ sí hay intercambio

$A[4]$ es el mayor de los hijos de $A[2]$

$A[4] < A[9]$ $(07 < 35)$ sí hay intercambio

$A[9]$ es el mayor de los hijos de $A[4]$

Luego de eliminar la segunda raíz, el montículo queda así:

56 44 36 35 21 28 12 15 07 16 13 08 27 10 60 67

Tercera eliminación

TERCERA ELIMINACIÓN DE LA RAÍZ

Se intercambia la raíz, 56, con el elemento que ocupa la última posición del montículo, 10. Las comparaciones que se realizan son:

$A[1] < A[2]$ $(10 < 44)$ sí hay intercambio
 $A[2]$ es el mayor de los hijos de $A[1]$

$A[2] < A[4]$ $(10 < 35)$ sí hay intercambio
 $A[4]$ es el mayor de los hijos de $A[2]$

$A[4] < A[8]$ $(10 < 15)$ sí hay intercambio
 $A[8]$ es el mayor de los hijos de $A[4]$

Luego de eliminar la tercera raíz, el montículo queda así:

44 35 36 15 21 28 12 10 07 16 13 08 27 56 60 67

Tabla eliminación

Eliminación										Montículo							
4	36	35	28	15	21	27	12	10	07	16	13	08	44	56	60	67	
5	35	21	28	15	16	27	12	10	07	08	13	36	44	56	60	67	
6	28	21	27	15	16	13	12	10	07	08	35	36	44	56	60	67	
7	27	21	13	15	16	08	12	10	07	28	35	36	44	56	60	67	
8	21	16	13	15	07	08	12	10	27	28	35	36	44	56	60	67	
9	16	15	13	10	07	08	12	21	27	28	35	36	44	56	60	67	
10	15	12	13	10	07	08	16	21	27	28	35	36	44	56	60	67	
11	13	12	08	10	07	15	16	21	27	28	35	36	44	56	60	67	
12	12	10	08	07	13	15	16	21	27	28	35	36	44	56	60	67	
13	10	07	08	12	13	15	16	21	27	28	35	36	44	56	60	67	
14	08	07	10	12	13	15	16	21	27	28	35	36	44	56	60	67	
15	07	08	10	12	13	15	16	21	27	28	35	36	44	56	60	67	

Algoritmo de eliminación

Eliminar_monticulo(A,N)

Enteros i, aux, izq, der, k, ap

Booleano bool

Para i = N hasta 2

 aux = A[i] , A[i]=A[1] , izq=2 , der=3, k=1 , bool=verdadero

 Mientras ((izq<i) y (bool==verdadero)) repetir

 mayor = A[izq] , ap=izq

 Si ((mayor < A[der]) y (der !=i)) entonces

 mayor = A[der] , ap = der

 Fin_si

 Si (aux < mayor) entonces

 A[k] = A[ap] , k = ap

 sino

 bool = falso

 fin_si

 izq = k*2 , der = izq + 1

 Fin_mientras

 A[k] = aux

Fin_para

Algoritmo de Montículo

Montículo (A,N)

//El algoritmo ordena los elementos del arreglo utilizando el método del montículo. A es un
//arreglo unidimensional de N elementos

1. Insertar_monticulo(A,N)
2. Eliminar_monticulo(A,N)