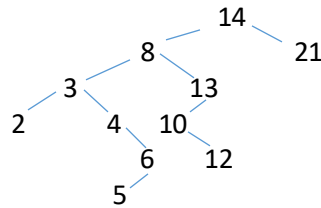


SOLEMNE 2 – SEMESTRE 2017-1 – SECCIÓN 554

Pregunta 1 [30 Puntos] Dada la siguiente lista: **14 – 8 – 21 – 3 – 4 – 6 – 5 – 13 – 2 – 10 – 12**

a) **[02 pts]** Insertar los datos en ese orden, en un árbol binario de búsqueda (ABB). Mostrar la traza de creación.

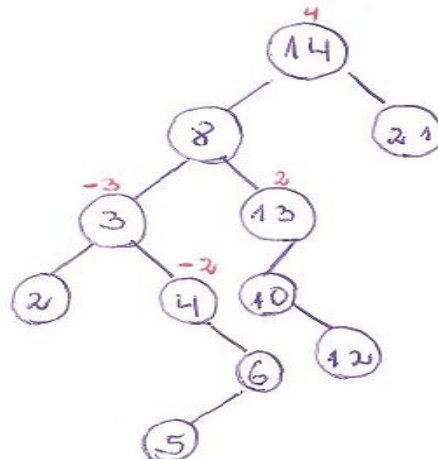


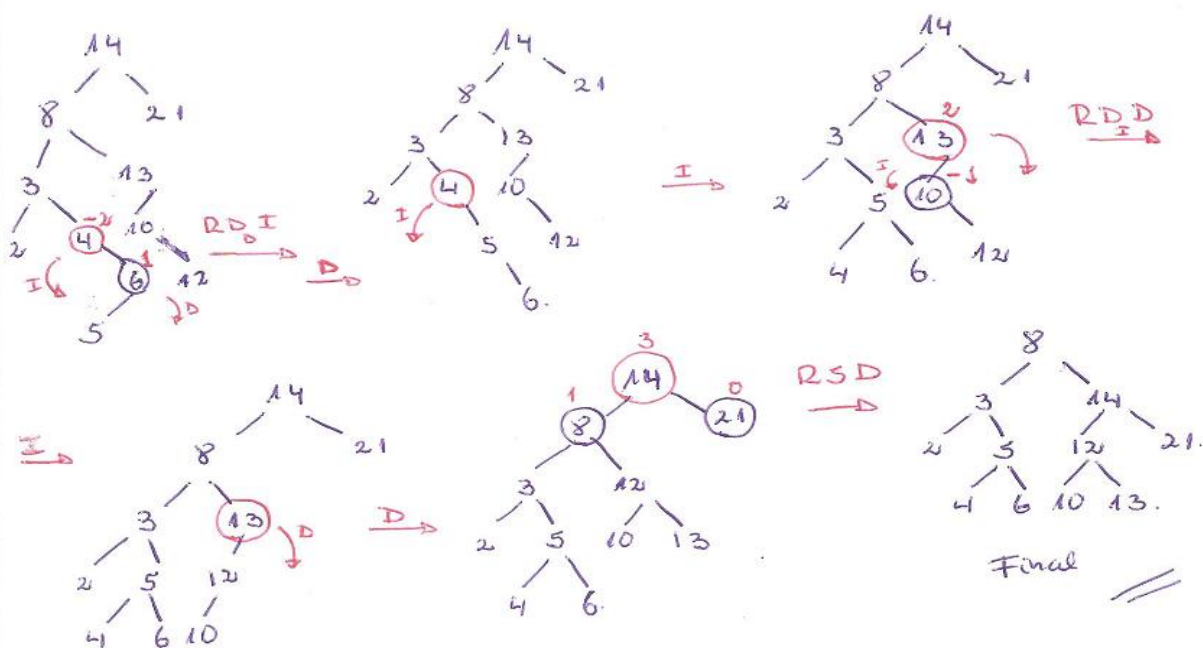
b) **[10 pts]** Responda y Justifique cada pregunta:

- i) Altura de 14: **5**, ya que la cantidad de aristas desde el nodo 14 a la hoja más lejana (nodo 5) cuentan 5.
- ii) Profundidad de 14: **0**, ya que el recorrido desde el nodo 14 hasta la raíz es 0
- iii) Longitud del camino entre 8 y 12: **3**, ya que las aristas del nodo 8 al 12 cuentan 3
- iv) Altura de 4: **2**, ya que la cantidad de aristas desde el nodo 2 a la hoja más lejana (nodo 5) cuentan 2.
- v) Profundidad de 4: **3**, ya que la cantidad de aristas desde el nodo 4 a la raíz (nodo 14) cuentan 3.
- vi) ¿Es completo?: **No**, no es completo ya que no se ingresaron todos los nodos en forma ordenada de derecha a izquierda. Ejemplo: se ingresaron los nodos hijos del nodo 3 sin haber ingresado previamente los hijos del nodo 21.
- vii) ¿es lleno?: **No**, no es lleno ya que no todos los nodos del árbol tienen sólo 2 hijos o ningún hijo. Ejemplo: el nodo 13 tiene un hijo.
- viii) Peso del árbol: **11**, ya que el peso de un árbol es la cantidad total de nodos.
- ix) Recorrido entre 8 y 6: **8-3-4-6**, ya que en el recorrido se indican los nodos desde un nodo inicial al final.
- x) Grado del árbol: **2**, ya que el grado de un árbol es la máxima cantidad de hijos que tienen los nodos del árbol.

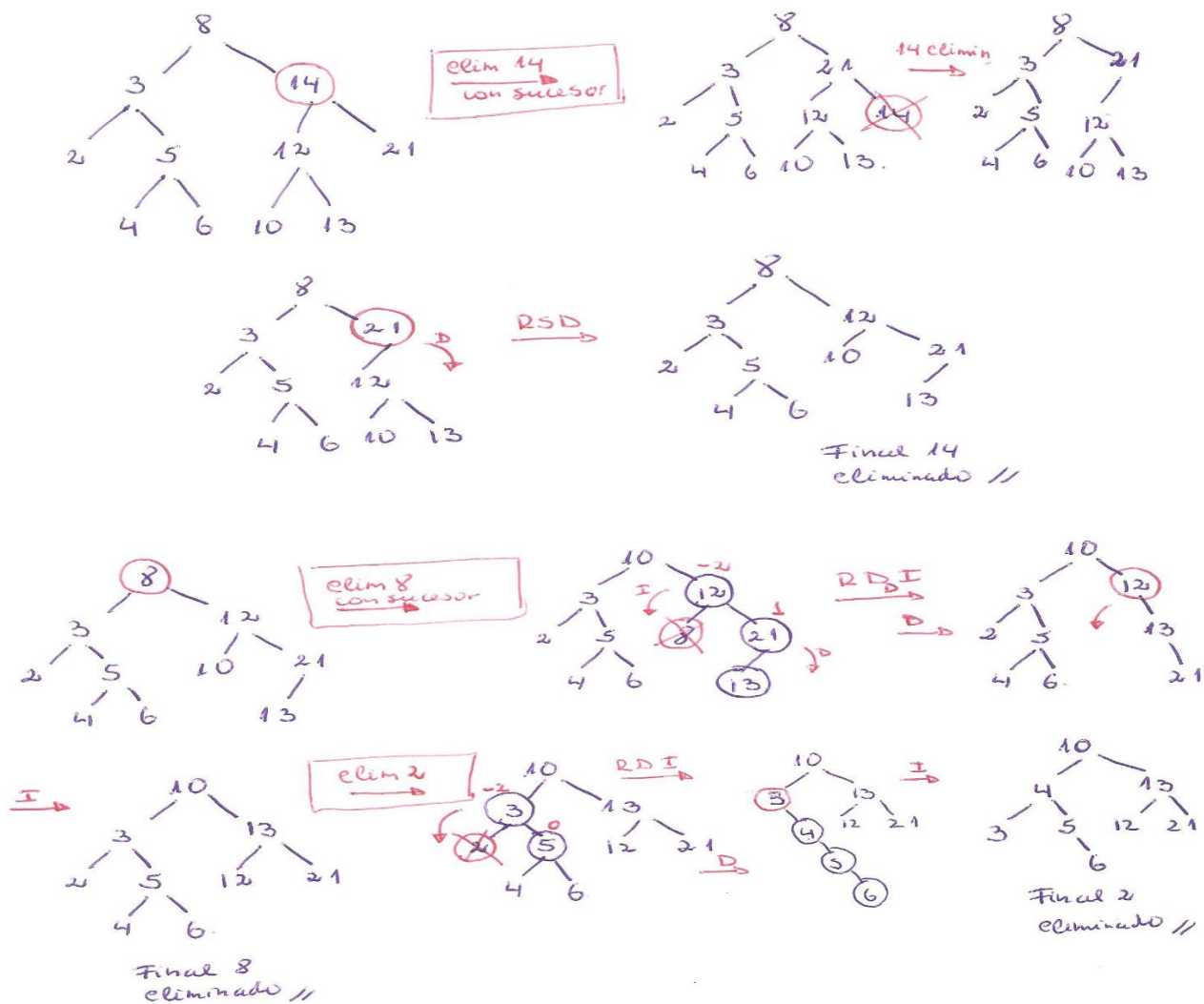
c) **[07 pts]** ¿El árbol es un AVL?. Fundamente su respuesta. En el caso que no sea realice la traza para transformarlo en AVL.

No, no es un árbol AVL ya que existen nodos en el árbol donde su Factor de Equilibrio (FE) es distinto a 0, 1 o -1, tal como se muestra en la figura siguiente:

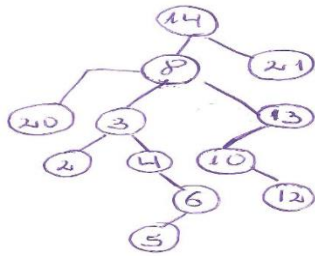




d) [07 pts] Muestre las operaciones necesarias (la traza), para eliminar los elementos 14, 8 y 2 en ese orden. Sólo cuando sea necesario utilice el sucesor y por cada operación debe conservar la propiedad de un árbol AVL.



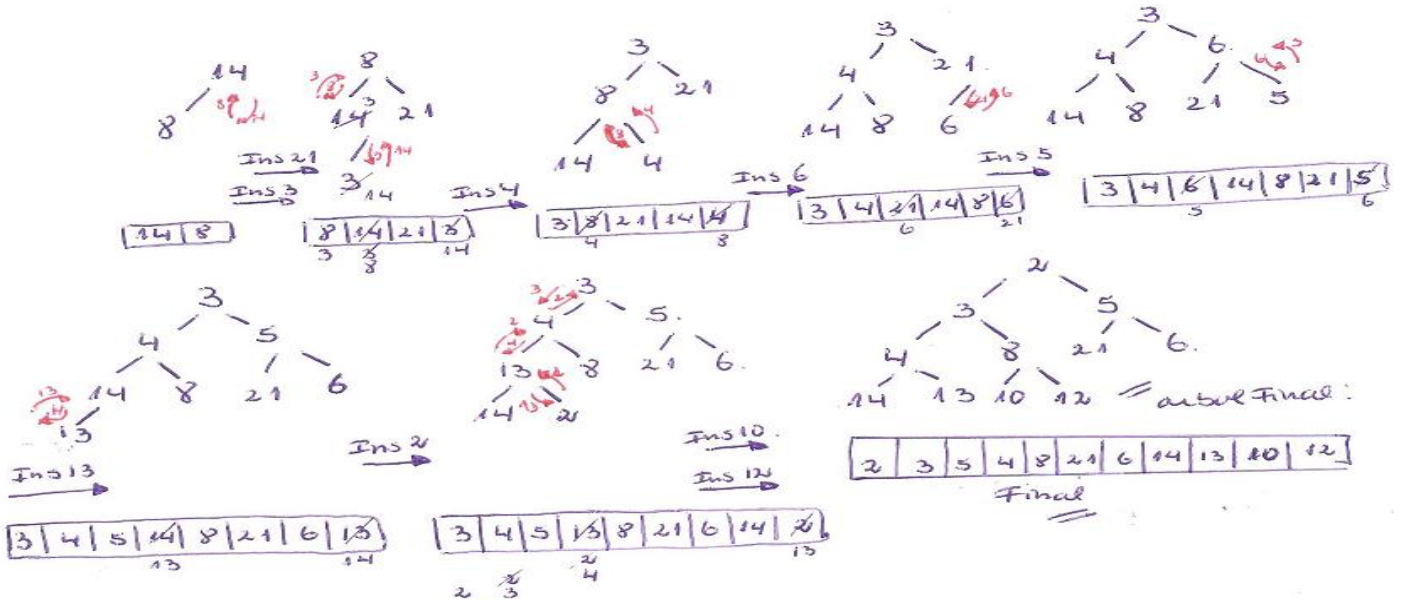
- e) [04 pts] Al árbol original inserte el hijo 20 al nodo 8. Insértelo al lado más izquierdo (árbol ya no es ABB). A este árbol realice el recorrido inorden y posorden.



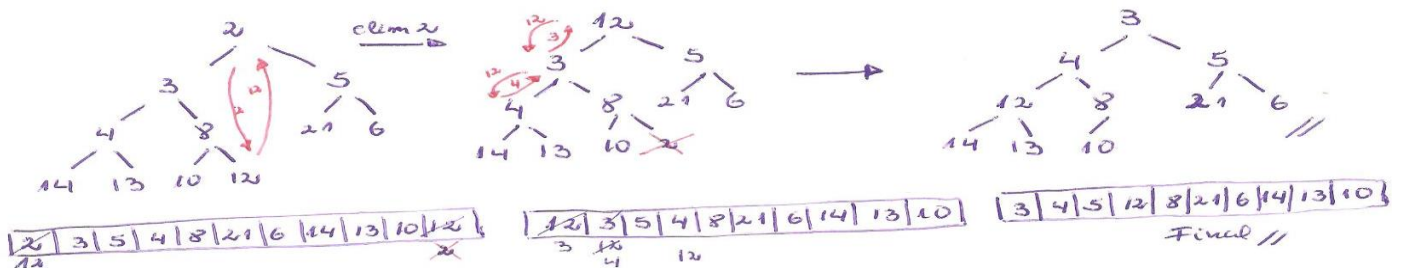
In-Orden 20 - 8 - 2 - 3 - 4 - 5 - 6 - 10 - 12 - 13 - 14 - 21
 Pos-Orden 20 - 2 - 5 - 6 - 4 - 3 - 12 - 10 - 13 - 8 - 21 - 14

Pregunta 2 [20 Puntos] Dada la siguiente lista: 14 - 8 - 21 - 3 - 4 - 6 - 5 - 13 - 2 - 10 - 12

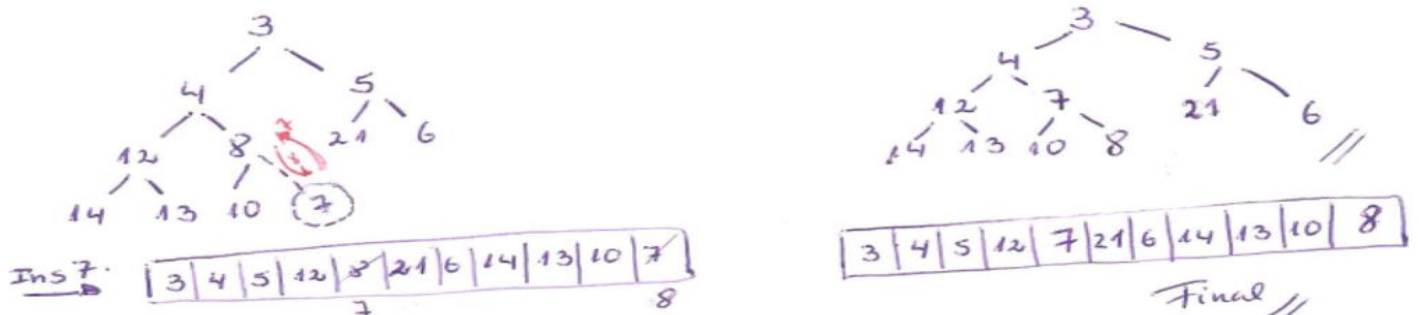
- a) [10 pts] Insertar los datos en un Min-Heap. Realice la traza de creación para el árbol y el arreglo asociado.



- b) [06 pts] Elimine un elemento del Heap. Realice la traza de eliminación para el árbol y el arreglo asociado.



- c) [04 pts] Inserte el número 7 al Heap. Realice la traza de inserción para el árbol y el arreglo asociado.



Pregunta 3 [20 Puntos]: Escriba una función que muestre (imprima) la altura de todos los nodos de un árbol dado. Asuma de que las funciones elementales, crear árbol, crear nodo e insertar nodo están implementadas. Se evaluará: **a)** La definición de los typedef; **b)** Los prototipos de todas las funciones utilizadas; **c)** La implementación del main (implemente todo lo necesario para la ejecución de lo solicitado. Utilice datos duros); **d)** Implementar las funciones NO asumidas como creadas; **e)** Implementar la función solicitada.

Definición de los typedef (2P)

Definición de los prototipos crear árbol, insertar nodo, imprimir alturas (4P)

Creación main (4p)

Crear función solicitada con los siguientes aspectos:

- **Recorrer cada nodo (2p)**
- **Calcular altura lado izquierdo (2p).**
- **Calcular altura lado derecho (2p)**
- **Comparar y asignar (2p)**
- **Función Base (2p)**

Pregunta 4 [30 Puntos]: Para un ABB dado, implemente una función que genere dos árboles binarios de búsqueda. El primero debe almacenar los números pares de los nodos de la sub-rama izquierda del árbol dado y el segundo debe almacenar las alturas distintas de 0 de los nodos de la sub-rama derecha del árbol dado. Asuma de que las funciones elementales, crear árbol, crear nodo, insertar nodo y eliminar nodo están implementadas.

Se evaluará: **a)** La definición de los typedef; **b)** La definición de los prototipos de todas las funciones utilizadas; **c)** La implementación del main (implemente todo lo necesario para la ejecución de lo solicitado. Utilice datos duros); **d)** Implementar las funciones NO asumidas como creadas; **e)** Implementar la función solicitada.

Definición de los typedef (2P)

Definición de los prototipos crear árbol, insertar nodo, eliminar nodo, nueva función (4P)

Creación main (4p)

Para sub-rama izquierda (recibida desde el main o de una función):

- **Recorrer cada nodo (2p)**
- **Para cada nodo Preguntar si almacena un número par(3p).**
- **Insertar árbol para número par encontrado (3p)**
- **Función Base (2p)**

Para sub-rama derecha (recibida desde el main o de una función):

- **Recorrer cada nodo (2p)**
- **Para cada nodo, llamar a la función que calcula altura del nodo(3p).**
- **Insertar árbol para altura distinta de 0 (3p)**
- **Función Base (2p)**

Puntaje	Nota	Puntaje	Nota	Puntaje	Nota	Puntaje	Nota
0.0	1.5	10.0	2.1	20.0	2.7	30.0	3.3
1.0	1.6	11.0	2.2	21.0	2.8	31.0	3.3
2.0	1.6	12.0	2.2	22.0	2.8	32.0	3.4
3.0	1.7	13.0	2.3	23.0	2.9	33.0	3.5
4.0	1.7	14.0	2.3	24.0	2.9	34.0	3.5
5.0	1.8	15.0	2.4	25.0	3.0	35.0	3.6
6.0	1.9	16.0	2.5	26.0	3.0	36.0	3.6
7.0	1.9	17.0	2.5	27.0	3.1	37.0	3.7
8.0	2.0	18.0	2.6	28.0	3.2	38.0	3.8
9.0	2.0	19.0	2.6	29.0	3.2	39.0	3.8

Puntaje	Nota	Puntaje	Nota	Puntaje	Nota	Puntaje	Nota
40.0	3.9	50.0	4.9	60.0	5.9	70.0	7.0
41.0	3.9	51.0	5.0	61.0	6.0		
42.0	4.0	52.0	5.1	62.0	6.1		
43.0	4.1	53.0	5.2	63.0	6.3		
44.0	4.2	54.0	5.3	64.0	6.4		
45.0	4.3	55.0	5.4	65.0	6.5		
46.0	4.4	56.0	5.5	66.0	6.6		
47.0	4.5	57.0	5.6	67.0	6.7		
48.0	4.6	58.0	5.7	68.0	6.8		
49.0	4.8	59.0	5.8	69.0	6.9		