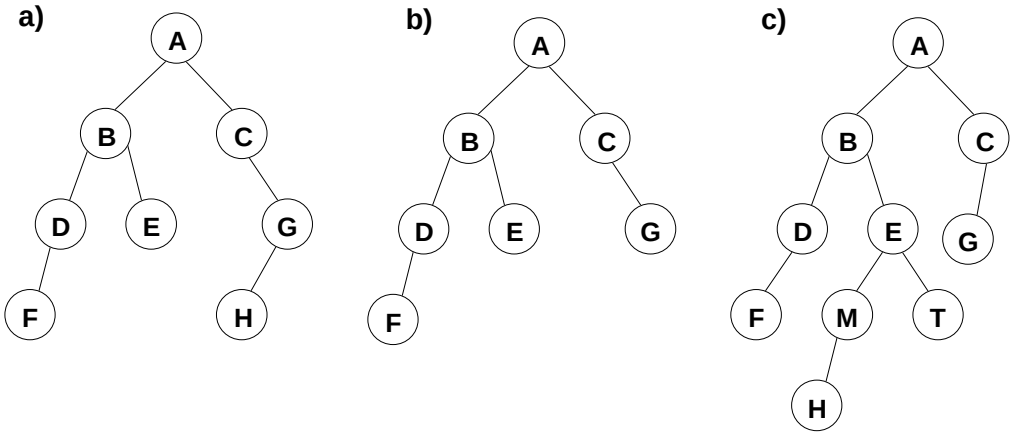
Carrera: Analista de Sistemas
Asignatura: Programación III
Docente: Lic. Verónica L. Vanoli

TRABAJO PRACTICO N°2
Tema: Arboles AVL

1. Determine si los siguientes árboles binarios son árboles AVL.

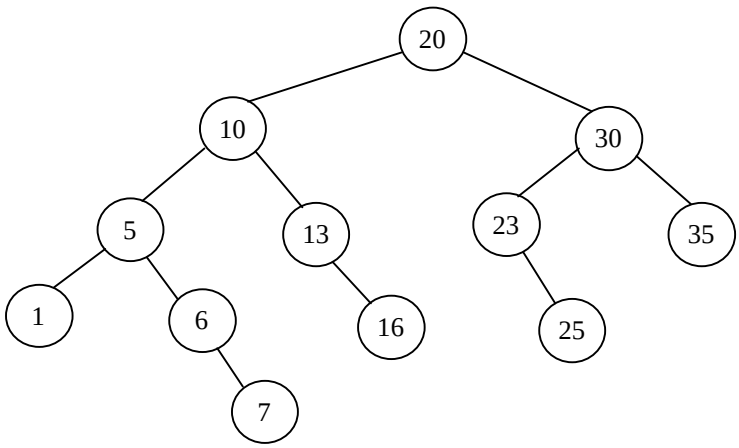


2. Dibujar el árbol AVL que resulta de insertar las claves 14, 6, 24, 35, 59, 17, 21, 32, 4, 7, 15, 22 partiendo de un árbol vacío.

3. Dada la secuencia de claves enteras: 100, 29, 71, 82, 48, 39, 101, 22, 46, 17, 3, 20, 25, 25, 10. Dibujar el árbol AVL correspondiente. Eliminar claves consecutivamente hasta encontrar un desequilibrio cuya restauración sea del tipo <<rotación simple>>, <<rotación doble>>.

4. Mostrar un ejemplo de árbol AVL donde al eliminar un elemento dado se requiera más de un rebalanceo para reequilibrar el árbol. Mostrar el árbol antes y después de la eliminación. (¡Ojo!, no se pide un ejemplo de rotación doble, sino, de más de un rebalanceo).

5. Dada la siguiente estructura de árbol, comprobar si se trata de un árbol AVL. En caso afirmativo, ¿cómo se puede comprobar si se trata del peor caso de esta estructura (en cuanto a máximo desequilibrio)?. En caso contrario, ¿cómo se puede reequilibrar para convertirlo en un árbol AVL?.



TRABAJO PRACTICO N°2
Tema: Arboles AVL

6. Se quiere leer un archivo de texto y almacenar en memoria todas las palabras de dicho texto y su frecuencia. La estructura elegida es un árbol AVL, ya que permite una búsqueda eficiente, independientemente de la entrada de los datos. Luego que la información del archivo esta almacenada en memoria, utilizando la estructura indicada, el programa permite obtener la cantidad de veces que una palabra, ingresada por teclado, se encuentra en el texto, si es que existe. Caso contrario la cantidad de veces es 0. (El programa debe manejar correctamente las mayúsculas y minúsculas, ya que río es la misma palabra que Río y que RIO).

7. Las N localidades de la provincia de Santa Cruz están dispuestas en un archivo. El archivo contiene el nombre de la localidad y la cantidad de habitantes.

a) Escribir un programa para disponer en memoria las localidades de la provincia de la siguiente forma: en un vector almacenamos las localidades, de manera tal que, cada elemento del vector tiene el nombre de la localidad y la raíz de un árbol AVL con los nombres de los habitantes de cada localidad. Al no estar los habitantes almacenados en el archivo, se tiene que solicitar su carga por teclado al usuario.

b) Se desea añadir la opción de cambiar el nombre de una persona de una determinada localidad. Habrá que tener en cuenta que esto puede suponer que ya no sea un árbol de búsqueda, por ello será necesario eliminar el nodo y después crear otro nuevo con la modificación introducida.

c) Se quiere que tenga la opción de que dado el pueblo A y el pueblo B, se elimine el pueblo A añadiendo antes sus habitantes al pueblo B.

8. Una empresa de servicios tiene tres departamentos: comercial (1), explotación (2) y comunicaciones (3). Cada empleado esta adscrito a uno de ellos. Se ha realizado una redistribución del personal entre los departamentos. El archivo Empresa contiene en el Num-ident-empleado, Origen y Destino. Origen toma los valores 1, 2 o 3 dependiendo del departamento al que pertenece el empleado. Destino toma los mismos valores, dependiendo del nuevo departamento asignado al empleado. El archivo esta desordenado. Escribir un programa que almacene el contenido del archivo Empresa en tres árboles AVL, uno por cada departamento Origen, y realice la redistribución en los árboles según el Destino.

9. Queremos construir una variante intermedia entre los árboles AVL y los árboles binarios de búsqueda no balanceados. Para ello, definimos los árboles BWM igual que los árboles AVL, pero la condición de balanceo es que la altura de los subárboles de cada nodo puede diferir como máximo en 2. Exponer las principales ventajas e inconvenientes de los árboles BWM respecto a los AVL. Mostrar el peor caso (número mínimo de nodos) de árbol BWM para altura 5.