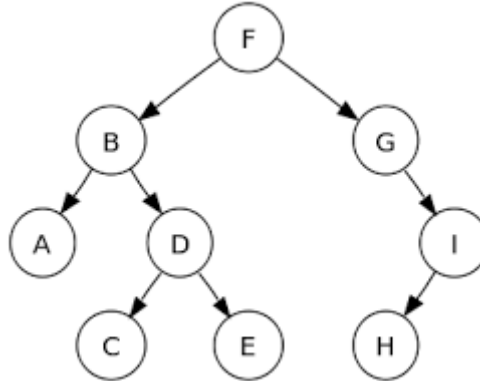


2° PARCIAL DE ALGORITMICA Y ESTRUCTURA DE DATOS II - 2017

1) Dadas dos listas (List) que contienen palabras ordenadas alfabéticamente, crear una tercer lista que contengan todas las palabras de las dos listas, ordenados alfabéticamente sin incluir las palabras repetidas.

2) Dado el siguiente árbol binario:



- a) Utilizar el TAD de árbol binario para crear el mismo.
- b) Recorrer el árbol y mostrar todos los nodos que no son hojas.
- c) Recorrer el árbol y mostrar todos los nodos que tienen un solo hijo.

3) Agregar al TAD de árbol binario el método **ancestor** que dado un nodo retorna una lista de todos sus nodos ancestros hasta la raíz.

Iterable<Position<E>> ancestor(Position<E> p)

Realizar un ejemplo que invoque el método implementado utilizando el árbol creado en el ejercicio anterior.

4) Desarrollar una aplicación que cargue alumnos en un árbol binario ordenado. De cada alumno se registra su número de legajo y su nombre. Cargar los mismos ordenados por legajo. Recorrer el mismo y cargar los datos de los alumnos en otro árbol ordenado por nombre.

TEORIA (Entregar en una hoja aparte)

Teoría

1. En el análisis de algoritmos, qué significa el estudio experimental y que deficiencias presenta. ¿Cuál es la propuesta superadora a este tipo de análisis? justifique.
2. ¿Qué es un Tipo Abstracto de Datos (TAD) y su interfaz? Explique qué significa la implementación de un TAD en lenguaje JAVA.
3. ¿Qué es un iterador; y cuál es la ventaja que una implementación de TAD lo defina?
4. ¿Qué casos se deben tener en cuenta al programar el método de eliminación del nodo de un árbol binario de búsqueda?