

El objetivo de esta sesión es desarrollar diferentes funciones de un árbol de búsqueda binaria, y su implementación en Python. La fecha límite de entrega de esta sesión es el mismo día de clase. Documentar el código.

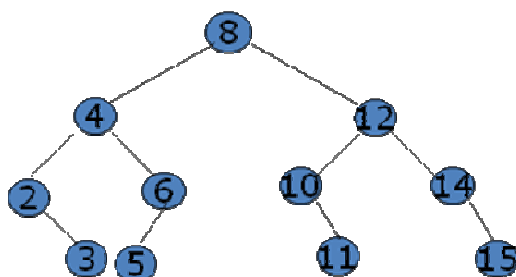
Ejercicio

En esta sesión vamos a continuar con la aplicación de LastFM. En este caso queremos representar a los usuarios en un árbol de búsqueda binaria.

Para implementar la aplicación seguir las instrucciones:

Paso 1. Crear un fichero **ABB_nombre_apellido.py** donde se definirán las siguientes operaciones de un árbol de búsqueda binaria:

- Un método para crear un ABB
- Un método para insertar un nuevo elemento dentro del árbol ABB. **Como clave para el árbol usaremos la relevancia**
- Un método para imprimir un árbol ABB en *postorden*
- Un método para calcular la ruta más larga de un árbol ABB. En caso de empate en el camino más largo, se quedará con la primera que encuentre. Por ejemplo, en el árbol descrito a continuación, la ruta más larga es: [8, 4, 2, 3]



- Un método para extraer todas las hojas del árbol. Por ejemplo, en el árbol anterior, el resultado sería [3, 5, 11, 15]

Paso 2. Crear un fichero nombre_apellido.py y leer el fichero LastFM_small.dat y guardarlo en un árbol de búsqueda binaria.

Paso 3. En el mismo fichero nombre_apellido.py, implementar un pequeño programa principal donde:

- Se cree e inicialice el árbol ABB
- Se carguen los datos del fichero en el árbol ABB creado anteriormente
- Imprimir por pantalla el árbol ABB usando la función definida en ABB_nombre_apellido.py
- Mostrar por pantalla la ruta más larga del árbol ABB
- Mostrar por pantalla las hojas del árbol ABB

NOTA: La entrega de la sesión ha de ser únicamente por el Campus Virtual dentro de las dos horas de clase.