Universidad de los Andes

Ingeniería de Sistemas y Computación
ISIS1206 – Estructura de Datos
Taller 7

Taller 7: Arboles Binarios

Operaciones sobre arboles binarios

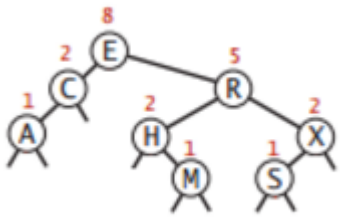
Objetivos

- Practicar la construcción de árboles dadas ciertas restricciones
- Aprender uno de los usos del árbol binario

Descripción General

En el texto guía del curso encontrará la información sobre diferentes tipos de recorrido que se pueden aplicar en un árbol binario. Es importante notar que diferentes árboles pueden resultar en la misma cadena de elementos al imprimirlos con el mismo tipo de recorrido.

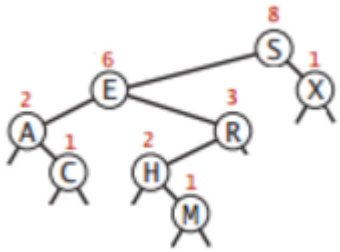
Por ejemplo, las llaves de este árbol



Ordenadas con el recorrido de in-orden, luce así:

A, C, E, H, M, R, S, X

Y las de este otro árbol:



Lucen igual:

A, C, E, H, M, R, S, X

Para determinar cómo era el orden original del árbol, sería necesario saber el orden en el que se agregaron los elementos, o en su defecto, conocer también la cadena resultante de un recorrido de pre-orden. En este taller se desea crear un algoritmo que dadas dos listas que definen el in-orden y pre-orden de un árbol, retorne el original donde se debe tener en cuenta que ninguno de los elementos del árbol está repetido.

Lo que usted debe hacer

Parte 1 – Preparación (en casa)

1. La aplicación debe leer la información de un archivo *.properties* que se encuentre en la carpeta data del esqueleto del taller, así pues debe crear el método que lea este archivo y almacene el pre-orden y el in-orden del árbol a construir.
2. Cree una estructura de datos de árbol binario simple donde pueda asignar los nodos izquierdo y derecho a su conveniencia.
3. Ahora, debe crear el método que reconstruya el árbol, utilice su estructura de datos para guardarlo. En el problema se asume que el pre-orden y in-orden son atributos del mundo. No tenga en cuenta casos en los que sea imposible reconstruir el árbol (por ejemplo, que sean recorridos de árboles diferentes). Asuma que el in-orden y pre-orden son correctos y pertenecen al mismo árbol.
4. Una vez tenga el árbol, debe crear el método que imprime el árbol en formato JSON; cada nodo debe tener un valor y si existen, un hijo derecho y un hijo izquierdo.
5. Por último, cree el método *crearArchivo* para generar el archivo de texto con el árbol.
6. Ejecute la aplicación y cargue el archivo “ejemplo.properties”.

```
-----------------------------------------
-                                       -
-           Siembra de árboles          -
-                                       -
-----------------------------------------
EL sistema para la plantación de árboles binarios

Menú principal:
-----------------
1. Cargar archivo con semillas
2. Salir

Seleccione una opción:
```

Figura 1 Menú principal

```
Recuerde que el archivo a cargar
debe ser un archivo properties
que tenga la propiedad in-orden,
la propiedad pre-orden (donde los
elementos estén separados por comas) y
que esté guardado en la carpeta data.

Introduzca el nombre del archivo:
ejemplo.properties
```

Figura 2 Carga del archivo .properties

7. Una vez terminada la ejecución abra el archivo JSON.

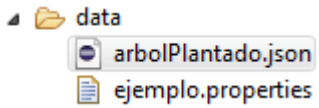


Figura 3 Ejemplo de Árbol (JSON)

8. Copie el contenido del archivo creado y visualícelo en http://codebeautify.org/jsonviewer.

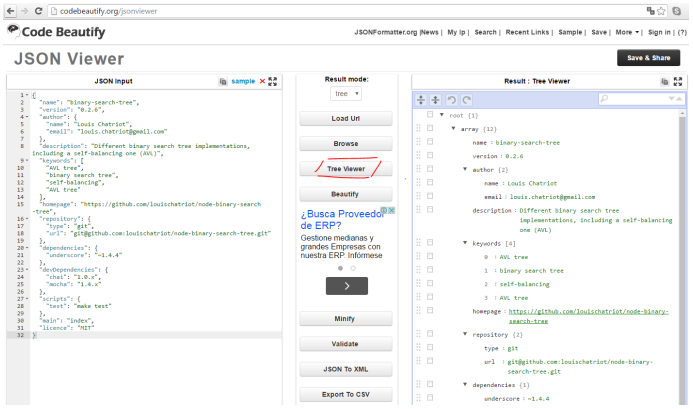


Figura 4 Visualizador de archivos JSON (ejemplo)

Parte 2 – Trabajo en clase

1. ¿Es posible lograr el mismo resultado con el pre-orden y pos-orden? Responda en el archivo README.md.
2. ¿Es posible lograr el mismo resultado con el in-orden y pos-orden? Responda en el archivo README.md.
3. Implemente un algoritmo que detecte si un árbol contiene un subárbol dado.
4. Cree otros ejemplos del archivo .properties con el fin de determinar si su aplicación funciona de forma apropiada. En particular se espera que cree un árbol y un subárbol con su nombre y apellido.

5. Realice pruebas unitarias para la aplicación de siembra de árboles las cuales se deben encontrar en el paquete taller.test. Como mínimo sus pruebas deberían validar los casos de la figura 5, y otras pruebas que considere pertinentes.

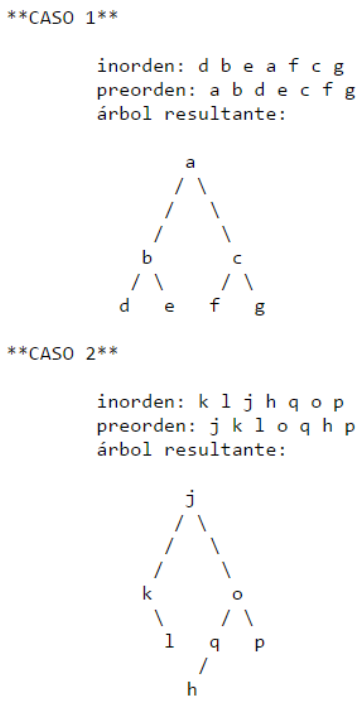


Figura 5 Casos de pruebas unitarias

Entrega

1. Verifique que su proyecto cumple con los requisitos de la entrega de talleres.
2. Responda las preguntas en el archivo README.md
3. Vuelva a correr las pruebas unitarias asegurándose que éstas evalúan casos críticos y reflejan el correcto funcionamiento de su entrega.
4. Entregue su taller por medio de **BitBucket**. Recuerde, si su repositorio no tiene el taller o está vacío, su taller no será calificado por más de que lo haya desarrollado.