

## Sesión 5 de Estructura de Datos:

Curso 2011/2012

El objetivo de esta sesión es conocer la estructura de árbol búsqueda binario, y su implementación en Python. La fecha límite de entrega de esta sesión es el día 06-05-2012. Documentar el código.

### Ejercicio

En esta sesión vamos a continuar con la aplicación del videoclub. En este caso queremos representar las películas mediante arboles de búsqueda binaria (ABB). La idea de la práctica es comparar los árboles de búsqueda binaria con las listas enlazadas.

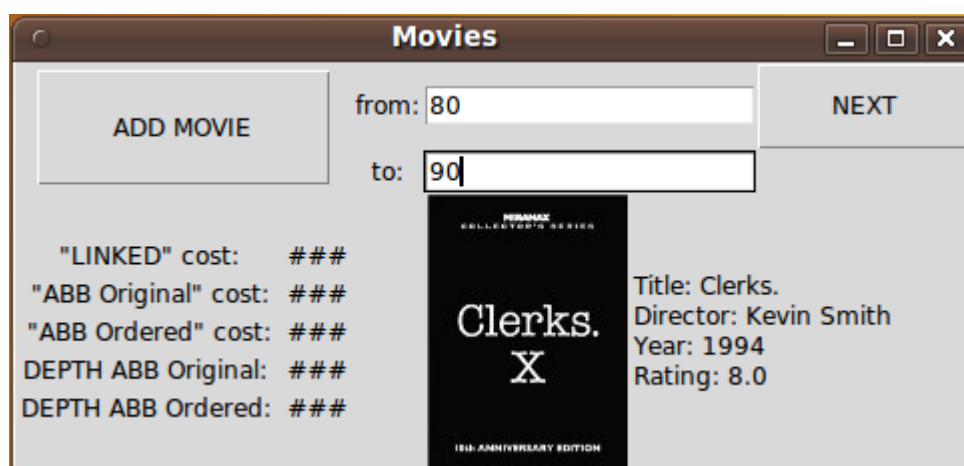
El primero de ellos será a partir de la lista **original** de nuestro archivo de películas.

La segunda manera será a partir de los datos del mismo archivo pero una vez estos estén ordenados en una lista según el valor de rating.

La tarea que deberéis efectuar en esta práctica es el análisis del tiempo necesario para la inserción y la búsqueda en las diversas estructuras de datos:

1. Lista enlazada
2. Árbol de búsqueda binaria "original" (ABB Original).
3. Árbol de búsqueda binaria "ordenado" (ABB Ordenado).

Todas estas funciones deben ser implementadas en la interficie TkInter del videoclub. En la siguiente figura podemos ver un ejemplo de la aplicación que debéis desarrollar.



### instrucciones:

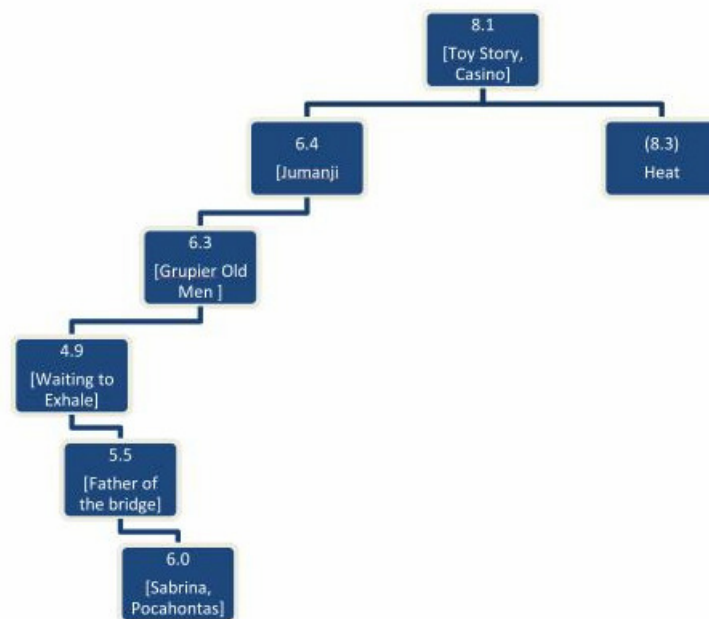
**Paso 1.** Crear dos árboles de búsqueda binaria: *ABB Original* y *ABB Ordenado*. Además, utilizando el código de la practica anterior, se ha de crear también una lista enlazada donde vamos a almacenar la películas en orden creciente de valor de rating. La lista enlazada y los árboles deben ser creados al apretar el botón ADD MOVIE.

El primer árbol (ABB Original) debe ser creado a partir de la lista de películas en el mismo orden que aparece en el archivo *.dat*. El segundo árbol a implementar (ABB Ordered) debe ser creado a partir de las lista de película que leéis del archivo *.dat* una vez esta esté ordenada según el valor en orden creciente de rating.

Tened en cuenta que cada nodo del árbol incluye una lista de películas con el mismo rating; de este modo, al añadir una nueva película debemos mirar si el nodo con el rating correspondiente existe.

El funcionamiento deberá ser el siguiente: en el caso de no existir debemos crear el nodo y añadir la película correspondiente en la lista de este nodo, y en caso de que el nodo ya exista únicamente debemos añadir la película en la lista del nodo del ABB que le corresponda según su rating.

En la siguiente figura tenéis un ejemplo de cómo quedaría el árbol con la entrada de datos más abajo.



Notad que para la creación de éste árbol los datos han sido introducidos con el siguiente orden:



**Paso 2.** Calcular el tiempo de inserción de las películas en los dos ABBs. Los dos tiempos se han de calcular al apretar el botón ADD MOVIE y han de ser enseñados

visualmente en la interfície como “ABB Original cost” y “ABB Ordered cost”. En la interfície también debéis enseñar la profundidad de los dos árboles implementados así como se puede ver en el ejemplo de la interfície (DEPTH ABB Original y DEPTH ABB Ordered).

**Paso 3.** Analizar y comparar el tiempo de búsqueda de un conjunto de películas utilizando las distintas estructuras de datos implementadas: Listas Enlazada, ABB Original y ABB Ordenado.

El funcionamiento es el siguiente:

- el usuario debe introducir en el campo *from* la puntuación mínima de rating y en el campo *to* el valor máximo de rating
- al apretar el botón *Next* el programa deberá calcular y enseñar en la interfície la **información de las películas** y el **tiempo de ejecución** para la búsqueda efectuada en las tres correspondientes estructuras de datos.

Tened en cuenta que cada vez que apretéis NEXT la aplicación debe mostrar la siguiente película (del resultado de la búsqueda efectuada) en orden ascendiente según su rating. Únicamente en el caso de modificar los valores de *from* y *to* la aplicación debe efectuar otra vez la búsqueda.

**Paso 4.** Estructurar bien la información, documentarla y argumentarla.

**NOTA:** La entrega de la sesión ha de ser únicamente por el Campus Virtual dentro del límite de tiempo predefinido.