

Estructuras de Datos

Curso 2012-2013. Convocatoria de Septiembre

Grado en Ingeniería Informática.

1. (2 puntos) Un videoclub está desarrollando una aplicación que permita un acceso rápido a las películas de las que disponen. No se tiene claro qué estructura de datos elegir para almacenar los registros (compuestos por código, título, año...). La búsqueda se hace por el título de la película. Para la inserción y borrado de películas, se usa un código único asociado a cada película. Además se necesita una función para imprimir de forma ordenada todas las películas. Analizar la eficiencia de las 4 operaciones si se usa como estructura de datos: 1) vector ordenado, 2) lista ordenada, 3) ABB, 4) AVL y 5) Tablas hash. En función del resultado obtenido en el análisis de eficiencia realizado, decidir que estructura de datos resuelve mejor el problema.
2. (2 puntos) Para gestionar un documento se utiliza un TDA Documento. Este TDA tiene en su representación una tabla hash, en la que cada palabra del documento tiene asociada una lista ordenada con las posiciones en las que aparece la palabra en el mismo. Implementar una función:

del Documento: min_distancia(string pal1, string pal2)

que devuelva la distancia mínima en la que aparecen las palabras *pal1* y *pal2* en el documento. En la representación de la tabla hash se usa hash abierto.

Por ejemplo, si se tiene una lista de enteros con elementos repetidos, diseñar (usando el TDA Documento) una función a partir de ella una lista ordenada de listas, de forma que los elementos iguales se agrupen en la misma sublista. Por ejemplo si se tiene $\{1,1,1,1,2,2,3,3,4,4,5,5,5,6,6,7,7\}$ entonces **salida** $= \{\{1,1,1,1\}, \{2\}, \{3,3\}, \{4,4\}, \{5,5,5\}, \{6,6\}, \{7,7\}\}$.

Un árbol de búsqueda *k*-balanceado, se dice que es *k*-balanceado si, para cada nodo, la diferencia entre el número de nodos en el subárbol izquierdo y derecho no es mayor que *k*. Diseñar (usando el TDA Documento) una función que permita determinar si un árbol es *k*-balanceado.

Se define una matriz *M* de tipo *T* como *M* de enteros (se almacenan los datos por filas) y se define una función *sumar* que suma por columnas sobre los elementos pares de la matriz. Por ejemplo, si se tiene la matriz *M* con los siguientes datos:

5	4	3
2	1	2
9	6	2
8	9	1

Entonces:

sumar *M* =

sumar *M* = $\{M[0][1] + M[2][1], M[1][0] + M[3][0], M[0][2] + M[2][2], M[1][1] + M[3][1], M[0][0] + M[2][0], M[1][2] + M[3][2]\}$

sumar *M* = $\{6, 10, 11, 10, 5, 1\}$