

Práctica Individual 2 – Complejidad y Árboles

A resolver en clases de prácticas por el profesor/a (NO hay que incluirlos en la entrega):

1. Implementar de forma recursiva e iterativa un algoritmo para el cálculo de la potencia a^n , siendo a de tipo Double y n de tipo Integer. Analizar sus tiempos de ejecución.
2. Analizar los tiempos de ejecución de las versiones recursivas sin y con memoria, iterativa y haciendo uso de la potencia de matrices para el cálculo de los números de Fibonacci. Comparar según los resultados sean de tipo Double o BigInteger.
3. Analizar los tiempos de ejecución de las búsquedas lineal y binaria, usando en los casos que correspondan listas ordenadas o desordenadas.
4. Implemente una función booleana que, dados un árbol binario de caracteres y una lista de caracteres, determine si existe un camino en el árbol de la raíz a una hoja que sea igual a la lista.
5. Diseñe un algoritmo que dado un árbol n-ario $\text{Tree}\langle E \rangle$ y un predicado sobre E devuelva una lista $\text{List}\langle \text{Boolean} \rangle$ de forma que el elemento i -ésimo de la lista será “True” si todos los elementos del nivel i cumplen el predicado.

A resolver por los estudiantes (SÍ hay que incluirlos en la entrega):

1. Dada la siguiente definición recursiva de la función f (que toma como entrada un número entero positivo):

$$f(a) = \begin{cases} 10, & a < 6 \\ \text{Math.pow}(a, 3) * f(a - 1), & \text{en otro caso} \end{cases}$$

Analizar los tiempos de ejecución de la versión recursiva e iterativa de dicha función, comparando según los resultados sean de tipo Double o BigInteger.

2. Analizar el tiempo de ejecución del algoritmo Mergesort, el cual se encuentra implementado en el repositorio de la asignatura. Analizar la influencia del tamaño del umbral del caso base.
3. Dado un árbol binario de caracteres, diseñe un algoritmo que devuelva un booleano en función de si dicho árbol está equilibrado. Un árbol binario está equilibrado si: (1) es vacío, (2) es hoja, o (3) la diferencia de alturas de sus dos hijos es como mucho 1, y además ambos están equilibrados.

Proporcione una solución también para árboles n-arios. Un árbol n-ario está equilibrado si: (1) es vacío, (2) es hoja, o (3) la diferencia de alturas de todos sus hijos es como mucho 1, y además todos están equilibrados.

4. Dado un árbol binario de enteros, diseñe un algoritmo que devuelva una lista de caminos con aquellos caminos desde el nodo raíz hasta una hoja que cumplan que: la suma de las etiquetas de los nodos del camino sea divisible entre la longitud de dicho camino. Se considera que un camino se modela como una lista que incluye los enteros de las etiquetas que componen el camino por orden (de la raíz a la hoja). Proporcione una solución también para árboles n-arios.

SE PIDE:

- Ejercicios 1 y 2:
 - Implemente o use directamente los algoritmos que se indican en el enunciado, así como un programa que mida sus tiempos de ejecución para los tamaños y casos necesarios.
 - Analice la complejidad teórica y los tiempos de ejecución usando la metodología explicada en clase.
- Ejercicios 3 y 4: proporcione una solución recursiva.

DEBE REALIZAR SU ENTREGA EN 2 PARTES:

- Proyecto en eclipse con las soluciones en Java.
- Memoria de la práctica en un único archivo PDF, que debe contener:
 - Código realizado.
 - Análisis de los tiempos de ejecución, incluyendo la complejidad teórica y las gráficas correspondientes para los ejercicios 1 y 2.
 - Volcado de pantalla con los resultados obtenidos para las pruebas realizadas, incluyendo al menos los resultados obtenidos para los tests proporcionados para los ejercicios 3 y 4.