

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. Analizar los tiempos de ejecución de las versiones recursiva e iterativa para el cálculo del factorial. Comparar según los resultados sean de tipo Double o BigInteger.
2. Analizar el tiempo de ejecución del algoritmo Quicksort usando la bandera holandesa. Analizar la influencia del tamaño del umbral del caso base, en el que se usará la ordenación por inserción.
3. Dados un árbol binario de caracteres y un carácter, diseñe un algoritmo que devuelva una lista con todas las cadenas que se forman desde la raíz a una hoja no vacía, excluyendo aquellas cadenas que contengan dicho carácter. Proporcione una solución también para árboles n -arios.
4. Dado un árbol binario de cadena de caracteres, diseñe un algoritmo que devuelva cierto si se cumple que, para todo nodo, el número total de vocales contenidas en el subárbol izquierdo es igual al del subárbol derecho. 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 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 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.