

TRABAJO PRACTICO N° 4

RECURSION

Nota1: para cada clase hacer un programa que pruebe cada uno de los métodos que provee.

Nota2: documentar cada clase y utilizar Javadoc para generar la misma.

1. Realizar dos implementaciones distintas para calcular el factorial de un número, una en forma recursiva y otra en forma iterativa. El programa de prueba debe pedir el valor del factorial a calcular desde la línea de comando, si el usuario no ingresa un valor, lo pide desde la consola. Utilizar excepciones para el tratamiento de errores si el valor ingresado es menor a cero. Realizar una traza del algoritmo recursivo para un número dado. Indicar la complejidad de ambos algoritmos.

2. Realizar dos implementaciones distintas para realizar una búsqueda binaria sobre un arreglo, una en forma recursiva y otra en forma iterativa, utilizar una función de envoltura para llamar a la función de recursión con una firma limpia. Realizar trazas del algoritmo recursivo mostrando distintos casos. Indicar la complejidad de ambos algoritmos.

3. Realizar una función recursiva que calcule el total de bytes utilizados, cantidad de directorios y cantidad de archivos desde un directorio dado. Realizar una traza del algoritmo e indicar su complejidad.

4. Realizar dos implementaciones recursivas para calcular la potencia de un número. Optimizar la segunda implementación multiplicando los resultados parciales. Probar con distintos números y medir el tiempo en obtener el resultado de cada algoritmo. Realizar una traza de ambos algoritmos para un número dado. Indicar la complejidad de ambos algoritmos.

5. Realizar dos implementaciones recursivas para calcular Fibonacci de un número. Optimizar la segunda implementación realizando una única llamada recursiva dentro de la función. Probar con distintos números y medir el tiempo en obtener el resultado de cada algoritmo. Realizar una traza de ambos algoritmos para un número dado. Indicar la complejidad de ambos algoritmos.

6. Realizar una implementación recursiva utilizando el método de ordenamiento quicksort para ordenar un array de enteros. Realizar una traza del algoritmo e indicar su complejidad.

7. Realizar una implementación recursiva para resolver el juego de las *Torres de Hanoi*. El usuario puede elegir la cantidad de discos y el algoritmo debe mostrar cómo se desplazan los mismos. Realizar una traza del algoritmo e indicar su complejidad.