UNIDAD TEMÁTICA 6 – CLASIFICACIÓN PARTE I

PRACTICOS DOMICILIARIOS INDIVIDUALES - 3

EJERCICIO 1

El algoritmo de ***Quicksort*** es uno de los más eficientes que existen, y ampliamente utilizado en todos los tiempos desde su descubrimiento.

Revisa cuidadosamente el algoritmo publicado en la bibliografía y escribe un muy breve resumen que indique cuál es la condición que se cumple en todo momento durante la ejecución del algoritmo

EJERCICIO 2

Utilizando la versión publicada en la presentación de la Cátedra, y asumiendo una función "***encuentraPivote***" que seleccione el menor de los dos primeros elementos del conjunto a ordenar, muestra paso a paso cómo se va ordenando el siguiente conjunto de datos:

44 - 55 - 12 - 42 - 94 - 18 - 6 – 67

- ¿cuántas llamadas se realizan en total al método "***Quicksort***" (contando la inicial)
- ¿cuál es el máximo nivel de profundidad recursiva alcanzado?
- ¿cómo podrías medir este nivel en una implementación en JAVA?

EJERCICIO 3

El algoritmo "***Quicksort***" es considerado uno de los algoritmos más importantes del siglo (pasado, y hasta el presente). Tanto es así que se utiliza como algoritmo de ordenación básico por la mayoría de lenguajes modernos. Así lo hace por ejemplo JAVA para implementar la ordenación de la clase Array (método "sort").

JAVA sigue evolucionando, y también ha mejorado su implementación de este algoritmo.

En base al artículo que puedes leer en la siguiente dirección: http://java.dzone.com/articles/algorithm-week-quicksort-three , realiza un resumen de las características más importantes de esa nueva versión, incluyendo el orden del tiempo de ejecución para conjuntos de datos ordenados de diferentes formas, conjuntos con claves duplicadas (análisis de la estabilidad del algoritmo), etc.

EJERCICIO 4

1. Desarrolla el análisis detallado del orden del tiempo de ejecución del algoritmo ***Quicksort*** de la presentación de la Cátedra.
   - ¿cuál es el peor caso?
   - ¿cuál es la probabilidad de que se presente este peor caso, cuando la distribución de probabilidades de las posiciones de las claves en el conjunto es una distribución uniforme – (es decir, cada clave tiene la misma probabilidad de aparecer en cualquier posición del conjunto inicial, en forma independiente de la posición de otras claves)
2. ¿cómo se puede modificar el algoritmo para que, cuando el conjunto es pequeño, se cambie a usar métodos directos más sencillos?