Práctica No. 3 Análisis de Algoritmos Empírico Métodos de ordenamiento

Competencia: Identificar cómo intervienen los factores que afectan el desempeño de un algoritmo, ejecutándolo en los tres escenarios posibles para ayudar en la toma de decisiones basada en su eficiencia.

Un método de ordenación muy simple, pero no muy eficiente, de elementos x_o,x_1, x_2, x_3,... x_n en orden ascendente consiste en encontrar el elemento más pequeño para la primera posición, luego el segundo elemento más pequeño para la segunda posición, esto se hace compararando cada elemento del vector con el resto de los elementos, en cada ocasión que encuentre que están no están en orden, los elementos deben intercambiarse. Al terminar la primera iteración el elemento más pequeño estará en la primera posición del vector. En la siguiente iteración se comienza por el segundo elemento y al terminar en la posición 1 estará el segundo elemento más pequeño. Las iteraciones continúan hasta que se llega a la penúltima posición en la que se ordenan los últimos dos.

Iteración para primera posición
Inicia comparando el 7 con el 3, como el 3 es menor se intercambian, luego compara el 3 con el 1 y como el 1 es menor se intercambian, el 1 se compara con el 2 y como ya están ordenados no hay intercambios.

7	3	1	1	1
3	7	7	7	7
1	1	3	3	3
6	6	6	6	6
2	2	2	2	2

Iteración para segunda posición
En este iteración el 1 ya esta en la posición menor, por lo tanto se empieza a compara a partir del segundo elemento, el 7 se compara con el 3 y como 3 es menor se intercambian. Luego el 3 se compara con el 2 y como es mayor se intercambian

1	1	1	1
7	3	3	2
3	7	7	7
6	6	6	6
2	2	2	3

Iteración para tercera posición
Como ya estan ordenados las primeras dos posiciones, ahora se inicia por la tercera posición, el 7 se compara con el 6 y como es menor se intercambian. Luego se compara el 6 con el 3 y también se intercambian.

1	1	1
2	2	2
7	6	3
6	7	7
3	3	6

Iteración para cuarta posición
En la última interaciǿn se comparan los elementos restantes, 7 y 6 como 6 es menor, se intercambian. No es necesaria una iteraciǿn más porque el 7 ya no tiene elementos con qué comparar.

1	1
2	2
3	3
7	6
6	7

Diseñe e implemente el algoritmo en lenguaje C.
Utilizando inicialmente un arreglo de 10 cadenas, determine el tiempo que tarda el algoritmo con los siguientes datos:

a) El tiempo y la cantidad de iteraciones para el peor de los casos
b) El tiempo y la cantidad de iteraciones para el mejor de los casos
c) El tiempo y la cantidad de iteraciones para cualquier otro caso
e) ¿Qué pasa si todos los datos son iguales?
d) ¿Qué pasa al cambiar el tipo de datos del vector de cadenas a entero?
f) ¿Qué pasa al aumentar el tamaño del arreglo?
g) ¿Cómo afecta el tiempo de ejecución la carga del sistema? Compruebe abriendo varias aplicaciones simultáneamente.

Ahora implemente el algoritmo en lenguaje Java y determine nuevamente los incisos a-g.

- Las funciones deben presentar en ***todo momento en pantalla*** los datos sobre los que se está realizando las comparaciones
- Elabore y suba a moddle un reporte con la siguiente estructura:

    - Portada
    - Descripción del algoritmo
    - Fotografía de su diseño del algoritmo
    - Preguntas a-g y sus respuestas con capturas de pantalla.
    - Elabore una tabla comparativa con los resultados de ambas implementaciones
    - Conclusiones.
    - Códigos de ambas implementaciones