

Actividad práctica. Algoritmos de ordenamiento

ATENCIÓN: Subir este document con las respuestas en formato PDF!

Título	Análisis de algoritmos de ordenamiento
Aprendizaje esperado (objetivo)	El alumno demostrará su capacidad para programar diferentes algoritmos de ordenamiento y medir el tiempo de ejecución de los mismos bajo determinadas condiciones, así como analizar e interpretar los resultados obtenidos, comparando los diferentes algoritmos.
Instrucciones	<p>Utilizando programación genérica (templates) y sobrecarga de operadores en C++, programa una clase Sorter que incluya los siguientes métodos y atributos:</p> <p>Métodos:</p> <ul style="list-style-type: none">Selection sortBubble sortInsertion sortMerge sortQuick sort <p>Cada método debe recibir por referencia el vector a ordenar.</p> <p>Genere un arreglo de 100 000 números enteros de manera aleatoria.</p> <p>Realice el ordenamiento de una copia del arreglo inicial (debemos hacer copiar sino se intentará ordenar un arreglo ya ordenado)</p> <p>Mida el tiempo de ejecución de cada caso y complete las tablas que aparecen más adelante en este documento.</p> <p>Genere algunas gráficas (en Google Sheets) comparando los resultados de todos los algoritmos y sus tiempos de ejecución.</p> <p>Analice e interprete los resultados alcanzados.</p> <p>Realice una copia de este documento en Google Docs y complete las secciones indicadas más adelante.</p> <p>Suba a la plataforma Canvas el archivo con sus resultados.</p> <p>Suba a Github todos los códigos programados.</p>

	No se aceptan trabajos fuera de fecha ni por correo electrónico.
Lugar en que se llevará a cabo	Casa
Forma de trabajo	Individual
Recursos	Foros de información en Internet Wikipedia (http://www.wikipedia.org) Códigos de algoritmos vistos en la materia Computadora
Tiempo estimado	5 horas

Respuestas

Repositorio de GitHub:

<https://github.com/a01027251/sort.git>

Tablas con los resultados de las mediciones:

Algoritmos de ordenamiento												
Tabla de resultados a completar (tiempo en ms)												
Alg.	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	Media	Stdev
Selection sort	37.18	36.15	35.56	29.99	31.64	32.07	29.16	28.23	44.86	48.19	33.815	5.20878691
Bubble sort	99.66	103.1	99.11	87.78	88	86.57	84.65	83.21	92.71	105.46	90.355	8.15262091
Insertion sort	39.54	40.77	37.78	36.68	36.34	34.6	34.97	35.77	37.15	47.85	36.915	3.91350156
Merge Sort	76.77	76.41	85.29	79.47	99.9	87.24	115.42	89.03	86.34	92.56	86.34	11.7949783

Interpretación de los resultados:

El selection sort y el insertion sort son los 2 algoritmos de ordenamiento mas eficientes hasta el momento con un promedio de 33 y 36 segundos respectivamente. Otra ventaja del insertion sort es que es muy constante en los tiempos que tarda en ordenar.

El merge sort resulta ser el más lento de los algoritmos de ordenamiento y de igual forma tiene la desviación estándar mayor, haciendo de este un algoritmo inconsistente.

El bubble sort resulta ser el algoritmo más lento, pero es estable con una desviación estándar mínima.

Gráficas comparativas

