**Practica 1: Sorting**

Forma

Descripción generada automáticamente con confianza media

Nombre: Hugo Levick Alvarez Luna

Grado y grupo: 2B

Fecha: 25/02/2022

**Introducción**

En esta práctica se ponen a prueba dos métodos de ordenamiento muy comunes: Bubble y Selection. Se comparará la velocidad, así como la cantidad de pasos que tienen que pasar para llegar al final del ordenamiento.

**Desarrollo**

Para reproducir estas pruebas, hice uso de los algoritmos que detalla el libro *Data Structures & Algorithms with JavaScript* por Michael McMillan.

Texto

Descripción generada automáticamente

Primero se crea una clase llamada “CArray” a la cual se le necesita enviar como parámetro cuántos números querremos dentro de él al momento de su creación. La propiedad *dataStore* es el lugar donde se guardan los números mediante un array o vector, *pos* se usa dentro de algunos métodos para indicar una posición dentro del array. El método *insert* se usa para insertar un valor dentro del array *dataStore.* El método *clear* se usa para establecer todos los valores del array en 0. El método *setData* se usa para establecer valores aleatorios dentro del array *dataStore*. El método *swap* se usa para cambiar de lugar dos valores dentro de *dataStore.* Finalmente están los métodos *bubbleSort* y *selectionSort* los cuales ejecutan el método de ordenamiento solicitado en *dataStore*.

Captura de pantalla de computadora

Descripción generada automáticamente

BubbleSort: Esta forma de ordenar (cuadrática) lee dos números dentro del array y los compara para saber cuál es menor y si están desordenados los cambia de lugar uno con otro. Este método es muy fácil de implementar pero es también muy lento comparado con otros métodos o algoritmos.

|  |  |  |
| --- | --- | --- |
| Corrida | Tiempo (ms) | Pasos |
| 1 | 286 | 25110053 |
| 2 | 279 | 24821283 |
| 3 | 259 | 25278985 |
| 4 | 276 | 25023173 |
| 5 | 293 | 24891423 |
| 6 | 292 | 24815772 |
| 7 | 298 | 24760329 |
| 8 | 258 | 24869067 |
| 9 | 281 | 25334060 |
| 10 | 264 | 25175837 |
| Promedio | 278.6 | 25007998.2 |

Texto

Descripción generada automáticamente

SelectionSort: Esta forma de ordenar tiene dos ciclos (cuadrática) el ciclo exterior se mueve por cada elemento del array y el interior lo compara con todos los demás elementos. Si el elemento exterior es mayor al interior se cambian de lugar. Este método también es relativamente fácil de implementar y tiene una gran mejor a comparación de BubbleSort.

|  |  |  |
| --- | --- | --- |
| Corrida | Tiempo (ms) | Pasos |
| 1 | 112 | 86613 |
| 2 | 91 | 86008 |
| 3 | 67 | 87074 |
| 4 | 67 | 86376 |
| 5 | 68 | 86144 |
| 6 | 73 | 84958 |
| 7 | 75 | 83956 |
| 8 | 110 | 87026 |
| 9 | 84 | 84690 |
| 10 | 80 | 85754 |
| Promedio | 82.7 | 85859.9 |

**Conclusiones**

**¿Para qué son útiles los métodos de ordenamiento?**

Sirven para poder organizar los elementos de un array de diferentes formas, pudiendo elegir el que más se adapte a tus necesidades de complejidad y costo.

**¿A qué familia de funciones pertenecen los algoritmo estudiados?**

A las funciones cuadráticas

**¿Cual de los dos algoritmos es más eficiente?**

Selection Sort. Tanto en tiempo como en pasos.

**¿Cómo sabemos cual es más eficiente?**

Por la cantidad de tiempo y pasos que toma, entre menos, mejor.

**¿Cómo podemos verificar que los números han sido ordenados correctamente?**

Imprimiendo el array y verificando

**¿En qué consiste la mejora del bubble-mejorado?**

En no volver a comparar los elementos que ya han sido ordenados

**¿Existe una mejora parecida para el selection-?**

No, pero existen mejores algoritmos que Selection.

**¿Existen otros métodos de ordenamiento?**

Si, algunos ejemplos son MergeSort, InsertionSort, QuickSort entre otros.

**¿Los algortimos estudiados son iterativos o recursivos?**

Iterativos porque no se vuelven a llamar a sí mismos

**¿Cómo puedes aplicar los algoritmos de ordenamiento en el proyecto del módulo integrado?**

No aplica

**¿Podemos predecir el tiempo y el número de pasos requeridos para ordenar 1000 y 100,000 números?**

Si, midiendo O en base a n

**Tus comentarios adicionales**

Es interesante aprender los diferentes tipos de métodos de ordenamiento y cómo funcionan, así como comparar sus tiempos para ordenar. Espero poder aprender métodos más rápidos y eficientes en el futuro y aprovecharlos al máximo.