TRABAJO ESCRITO DEL PROYECTO

PROYECTO #1 - COMPLEJIDAD COMPUTACIONAL EN LOS ALGORITMOS DE ORDENAMIENTO

Profesor: Edgar Tista García

Asignatura: Estructura de Datos y Algoritmos II

Grupo: 5

Integrantes: , Zarate Menes Quetzalli

No. de lista: ,35

Semestre: 2024-2

Fecha de entrega: 24 de marzo del 2024

Observaciones:

CALIFICACIÓN:

OBJETIVO

Que el alumno observe la complejidad computacional de los algoritmos de ordenamiento para comparar su eficiencia de ejecución en grandes volúmenes de información

INTRODUCCIÓN

En nuestro proyecto, abordamos el estudio teórico y práctico de los algoritmos de ordenamiento, fundamentales en el campo de la ciencia de la computación. Estos algoritmos son esenciales para la organización y manejo eficiente de datos, una habilidad clave en la era de la información.

InsertionSort es un algoritmo intuitivo que construye la salida ordenada elemento por elemento, insertando cada nuevo elemento en su posición correcta dentro de la parte ya ordenada.

SelectionSort mejora este proceso seleccionando el elemento más pequeño de la parte no ordenada y colocándolo al principio de la lista ordenada.

Heapsort transforma la lista en un montículo binario y luego extrae los elementos de manera ordenada, aprovechando la estructura de árbol para mejorar el tiempo de ejecución.

Bubblesort es un método simple que revisa repetidamente la lista para intercambiar elementos adyacentes que están en el orden incorrecto.

Quicksort divide la lista en dos partes, ordenando cada una de ellas de manera independiente a través de un proceso recursivo basado en un pivote.

Mergesort también utiliza la recursividad, dividiendo la lista en mitades hasta llegar a listas de un solo elemento y luego combinándolas de manera ordenada.

Finalmente, exploraremos un algoritmo adicional, que seleccionaremos por su relevancia y eficacia comparativa, para ampliar nuestra comprensión de estas técnicas de ordenamiento.

Nuestro análisis se centrará en cómo estos algoritmos manejan el incremento en el tamaño de los datos, observando la cantidad de operaciones necesarias para alcanzar la lista ordenada final. Este enfoque nos permitirá no solo entender los principios básicos de funcionamiento de cada algoritmo, sino también su comportamiento y rendimiento en condiciones reales y variadas.

DESARROLLO

InsertionSort

SelectionSort

HeapSort

BubbleSort

QuickSort

MergeSort

Adicional-CountingSort

Menu Principal

IMPLEMENTACIÓN

CONCLUSIONES