Escuela de Ingeniería en Computación

Curso: IC-3002 Análisis de Algoritmos

Profesor: Ing. Joss Pecou Johnson

TEC | Tecnológico de Costa Rica

Tarea Corta #1: Implementación y Análisis de Algoritmos

Parte 1: Implementación y Análisis de Merge Sort

Objetivo: Implementar y analizar el algoritmo de ordenamiento Merge Sort, un algoritmo de ordenamiento basado en el paradigma de divide y vencerás.

Instrucciones:

1. Implementar el algoritmo de Merge Sort en el lenguaje de programación Python.
2. Analizar la complejidad temporal del algoritmo para el mejor caso, peor caso y caso medio.
3. Proporcionar la función T(n) que describe la complejidad temporal del algoritmo.
4. Aplicar el algoritmo para una lista de números aleatorios de al menos 100 elementos y presentar los resultados obtenidos en tiempo de ejecución. (Al menos 3 resultados)
5. Comparar la eficiencia de Merge Sort con otro algoritmo de ordenamiento de su elección.

Entrega:

- Código fuente de la implementación del algoritmo.
- Explicación del algoritmo.
- Análisis de complejidad temporal incluyendo la función T(n).
- Resultados del aplicar el algoritmo para diferentes casos.
- Comparación de eficiencia con otro algoritmo de ordenamiento.

Parte 2: Análisis de Búsqueda Binaria en una Lista de Listas

Objetivo: Analizar la eficiencia del algoritmo de búsqueda binaria aplicado a una lista de listas.

Instrucciones:

1. Implementar una función que realice una búsqueda binaria en una lista de listas. Cada lista interna debe estar ordenada, pero las listas internas pueden no estar ordenadas con respecto a otras listas.
2. Analizar la complejidad temporal del algoritmo implementado.
3. Proporcionar la función T(n) que describe la complejidad temporal del algoritmo.
4. Aplicar la implementación del algoritmo a una estructura de datos con al menos 10 listas internas, cada una con al menos 50 elementos. (No puede haber elementos repetidos)
5. Presentar los resultados obtenidos en términos de tiempo de ejecución y comparar la eficiencia del algoritmo con una búsqueda lineal tradicional en la misma estructura de datos.

Entrega:

- Código fuente de la implementación.
- Explicación del algoritmo.
- Análisis de complejidad temporal incluyendo la función T(n).
- Resultados del aplicar el algoritmo para diferentes casos.

- Comparación de eficiencia con la búsqueda lineal tradicional.

Formato de Entrega:

Entregar un dentro de una carpeta comprimida denominada 'Tarea_Corta1-nombre1_nombre2' con dos archivos .py cada uno con sus respectivos algoritmos, casos de prueba y documentación interna, además, un archivo pdf que contenga el análisis para ambos escenarios planteados.
Documento: El documento de pdf debe estructurarse de la siguiente manera:

- **Portada**
- **Parte 1: Implementación y Análisis de Merge Sort**
  - Descripción del algoritmo.
  - Análisis de complejidad.
  - Función T(n).
  - Resultados.
  - Comparación de eficiencia con otro algoritmo.
- **Parte 2: Análisis de Búsqueda Binaria en una Lista de Listas**
  - Descripción del algoritmo.
  - Análisis de complejidad.
  - Función T(n).
  - Resultados.
  - Comparación de eficiencia con otro algoritmo.

Criterios de Evaluación:

- Correctitud y eficiencia del código.
- Claridad y profundidad del análisis de complejidad.
- Calidad de los resultados experimentales y su presentación.
- Comparación y análisis crítico de la eficiencia de los algoritmos.
- Organización y presentación del documento.

Fecha de Entrega:

La tarea debe ser entregada el 21 de agosto del 2024 con hora límite de las 11:55 p.m., cualquier tarea presentada después de la hora establecida no será evaluada.