ANÁLISIS Y DISEÑO DE ALGORITMOS

Documentación Proyecto Final  
Huffman Code

MILTON DAVID ZAPATA AGUILERA

1866060-2711

JULIAN ANDRES IBAÑEZ QUIÑONES

1866221 - 2711

UNIVALLE

SEDE TULUÁ

DESARROLLO DE SOFTWARE II

15/12/2023

CONTENIDO

Pág.

[1. Implementación de la codificación Huffman 3](#_Toc153594551)

[A. Generación del árbol 3](#_Toc153594552)

[2. Análisis de Complejidad 4](#_Toc153594553)

[3. Ejemplos Y conclusiones 4](#_Toc153594554)

[A. Ejemplos 4](#_Toc153594555)

[D. Discusión Resultado. 6](#_Toc153594556)

# Implementación de la codificación Huffman

## Generación del árbol

Diccionario de Frecuencias (frecuencia):

* Estructura de Datos: Diccionario ({}).
* Propósito: Almacena la frecuencia de cada carácter en la cadena de entrada.

Lista de Nodos (nodos):

* Estructura de Datos: Lista ([]).
* Propósito: Almacena los nodos del árbol de Huffman junto con sus frecuencias. La lista se mantiene ordenada de manera descendente según las frecuencias.

Clase NodoArbol:

* Estructura de Datos: Clase.
* Propósito: Define la estructura de un nodo en el árbol de Huffman. Cada instancia de esta clase tiene referencias a sus nodos hijos izquierdo y derecho.

Función arbol\_codificacion\_huffman:

* Estructura de Datos: Recursión y un diccionario ({}).
* Propósito: Utiliza la recursión para construir un diccionario que representa los códigos Huffman asignados a cada carácter.

Diccionario de Códigos Huffman (codigo\_huffman):

* Estructura de Datos: Diccionario ({}).
* Propósito: Almacena los códigos Huffman asignados a cada carácter después de construir el árbol de Huffman.

# Análisis de Complejidad

# Ejemplos Y conclusiones

## Ejemplos

Ejemplo 1

Captura de pantalla de computadora

Descripción generada automáticamente

Ejemplo 2

Captura de pantalla de computadora

Descripción generada automáticamente

## D. Discusión Resultado.

Estos resultados indican que, en este caso específico, la codificación de Huffman y la serialización del árbol aumentaron el tamaño total del archivo en comparación con el archivo original. Esto puede ocurrir en casos donde la longitud de la cadena original es corta o cuando hay redundancia en los datos que no es bien manejada por el algoritmo de compresión.

Es importante tener en cuenta que la eficacia de la compresión de Huffman puede variar según la naturaleza de los datos. En situaciones donde hay repeticiones y patrones más evidentes, la compresión de Huffman tiende a ser más efectiva. En datos más aleatorios o con menos repeticiones, la compresión puede no ser tan eficiente.