UNIVERSIDAD DE GUADALAJARA

Centro Universitario de Ciencias Exactas e Ingenierías

División de Tecnologías para la Integración Ciber-Humana

Análisis de algoritmos

Jennifer Patricia Valencia Ignacio, Código: 223991721

Elizabeth Arroyo Moreno, Código: 221453749

Karla Rebeca Hernández Elizarrarás, Código: 223991977

Ingeniería en computación

Técnica Coraz Huffman

26 de Octubre de 2025

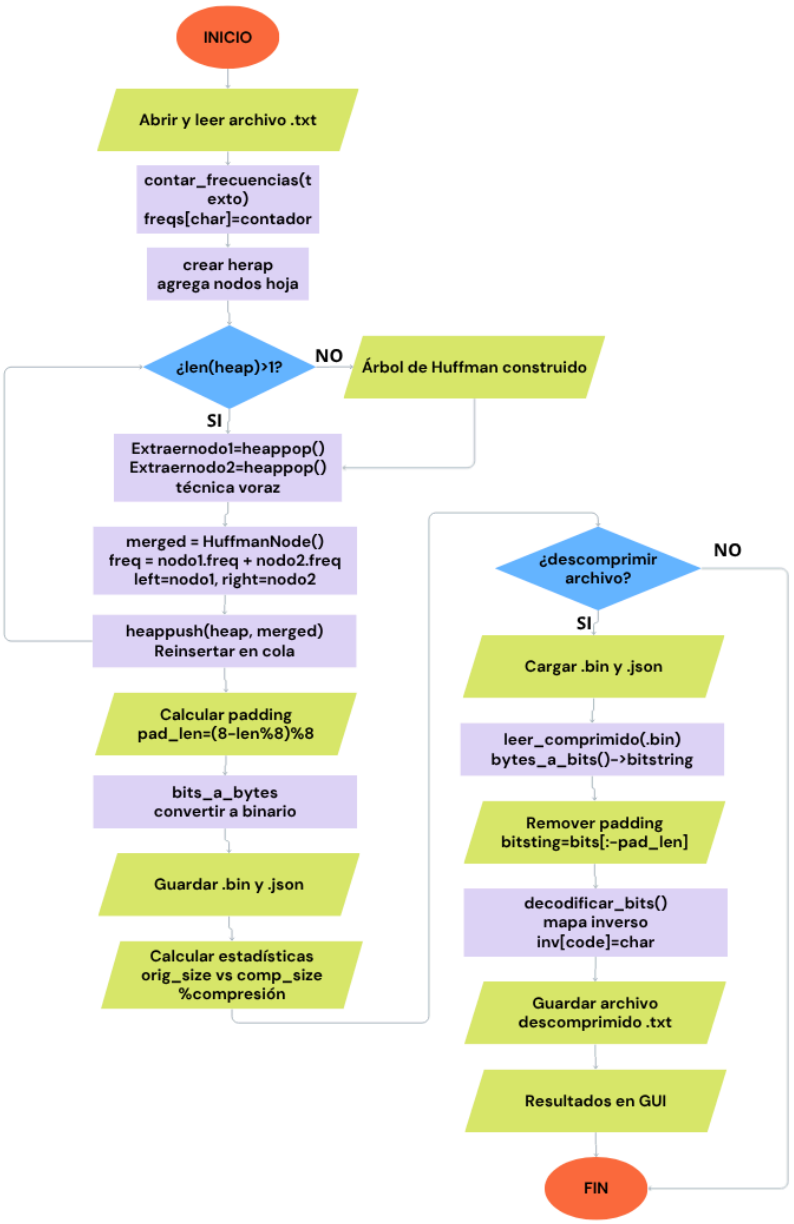
Introducción

El algoritmo de huffman sirve para comprimir archivos de texto reduciendo su tamaño sin perder información. Este algoritmo usa una técnica voraz que da códigos binarios más cortos a los caracteres más usados y códigos más largos a los menos frecuentes y así el archivo usa menos espacio.

Objetivo

El objetivo es leer un archivo de texto, calcular la frecuencia de cada carácter que tiene el texto y luego usar el algoritmo de Huffman para construir un árbol binario, A base de ese árbol, se generan códigos binarios para cada carácter, se codifica y decodifica el archivo y al final se comparan los tamaños del archivo original y el comprimido para ver cuánto cambió.

Desarrollo



El código se divide en varias partes:

1. Contar frecuencia de cada carácter
   Se recorre todo el texto y se guarda cuantas veces aparece cada carácter en un diccionario. Esta es la base para crear el árbol.

2. Crear árbol de Huffman
   Con las frecuencias se crean nodos, cada nodo con su carácter y frecuencia. Después, se van juntando los dos nodos con menos frecuencia y se repite hasta que solo quede uno, creando el árbol completo.

3. Generar código

Se recorre el árbol desde la raíz, cada vez que va a la izquierda se agrega un “0” y cada vez que va a la derecha se agrega un “1” y así cada carácter obtiene su código binario.

4. Codificar texto
   Ya que cada carácter tenga su código binario se reemplaza en el texto y todo eso se guarda como una cadena de bits y se escribe en un archivo .bin, que es el texto comprimido.

5. Decodificar texto
   Para recuperar el texto original, se lee el archivo .bin y se usa la tabla de códigos para volver a construir las letras originales. Esto sirve para comprobar que la compresión no dañe el texto.

6. Interfaz gráfica
   En la GUI hay 4 botones:
   - Abrir .txt: selecciona el archivo a comprimir
   - Comprimir: crea el .bin y el .json
   - Descomprimir: lee los archivos comprimidos y construye el texto original
   - Ver códigos: muestra la lista de códigos generados por Huffman.

Guía para utilizar el programa.

Para probar el algoritmo se deben seguir los siguientes pasos.

1. Abrir un Archivo txt.
   - Hacer clic en el botón que dice “Abrir .txt”
   - Seleccionar el archivo que deseas comprimir.

Una vez seleccionado se cargará el archivo y te avisará que todo está en orden mostrando lo siguiente: Nombre del archivo, el tamaño en bytes y vista previa de su contenido (los primeros 10,000 caracteres).

2. Comprimir un archivo.
   - Después de que el archivo se haya cargado exitosamente hacer clic en la opción de “Comprimir” .
   - Seleccionar la carpeta donde deseas guardarlo.
   - Al guardar automáticamente se generarán 2 archivos:
     - .bin (Archivo comprimido).
     - .json (Códigos de huffman).
   - Por medio de la GUI se mostrarán las estadísticas de los resultados.
     - Tamaño original.
     - Tamaño comprimido.
     - Ahorro en bytes y porcentaje.

3. Descomprimir un archivo.
    - Dar clic en “Descomprimir”.
    - Seleccionar el archivo .bin comprimido.
    - Seleccionar el archivo con los códigos huffman (.json).
    - Escoger la carpeta donde se quiere guardar el resultado.

Finalmente se generará el archivo descomprimido llamado “_decompressed.txt” con su contenido original.

4. Códigos Huffman.
    - Hacer clic en “Ver códigos”
    - En seguida se abrirá una nueva ventana mostrando cada carácter con su código binario de forma ordenada por longitud de código.

5. Guardar códigos.
    - Dar clic en “Guardar Códigos (.json)”
    - Elegir la ubicación del archivo .json y su nombre.
    - Al terminar se guardará el archivo.

Tabla de Asignación de Tareas

	Código	Reporte
Eli	Implementación del algoritmo de Huffman (técnica voraz) Función construir_arbol() usando heap/cola de prioridad Función generar_codigos() con recursión DFS Medición de tiempos de ejecución Validación de extracción correcta del árbol Testing de funciones core	Revisión y formato final del documento Explicación técnica de la técnica
Rebe	Implementación de procesamiento de archivos Función contar_frecuencias() - análisis de texto Función codificar_texto() y decodificar_bits() Funciones bits_a_bytes() y bytes_a_bits() Lógica de padding y conversión binaria Manejo de archivos .bin y .json Pruebas de integridad	Diagrama de Flujo: Creación completa del algoritmo Huffman Análisis de diferentes archivos de prueba

	compresión/descompresión	
Jenny	Implementación completa de GUI con Tkinter Clase HuffmanGUI - interfaz gráfica completa Integración de todos los módulos Sistema de carga/guardado de archivos Módulo de análisis estadístico y cálculo de compresión Visualización de resultados Prueba final de integración	Pseudocódigo: Desarrollo detallado del algoritmo Documentación de funciones de procesamiento

Conclusión

Se entendió como el algoritmo de Huffman puede comprimir archivos de forma eficiente usando una estrategia voraz. El programa funciona correctamente, puede comprimir y descomprimir archivos sin que se pierda información. Se logró el objetivo de aplicar el algoritmo, analizarlo y probar su eficiencia. Fue una buena forma de aprender como los árboles binarios pueden servir para algo práctico.

Referencias

de Huffman, C. de A. M. el A. (s/f). *Módulo 3 / Aplicación práctica*. Com.mx. Recuperado el 8 de noviembre de 2025, de https://libroweb.alfaomega.com.mx/book/393/free/data/Capitulos/cap16.pdf

W3Schools.com. (s. f.). https://www-w3schools-com.translate.goog/dsa/dsa_ref_huffman_coding.php?_x_tr_sl=en&_x_tr_tl=es&_x_tr_hl=es&_x_tr_pto=tc

Levitin, A. (2008). Introduction to design and analysis of algorithms, 2/E. Pearson Education India.

Wong, K. J. (2024, mayo 10). *Python tree implementation: A guide*. Built In. https://builtin.com/articles/tree-python

Extensión de archivo .bin: todo acerca de la extensión de archivo. (2023, marzo 1). IONOS Digital Guide. https://www.ionos.mx/digitalguide/servidores/know-how/extension-de-archivo-bin/

Erickson, J. (2024, abril 4). *¿Qué es JSON?* Oracle.com; Oracle. https://www.oracle.com/latam/database/what-is-json/