

Hoja de trabajo No. 9

Realizar: Compresor de Huffman.

Realizarse: En parejas.

Objetivos:

- Implementación de el algoritmo de huffman utilizando un árbol binario.
- Crear un compresor de archivos de texto plano

Programa a realizar:

En esta práctica, desarrollarás una aplicación que permita comprimir archivos de texto utilizando el algoritmo de codificación de Huffman. Este algoritmo consiste en construir un árbol binario de Huffman a partir de la frecuencia de los caracteres del archivo original, generando así un código binario óptimo para cada carácter.

El sistema debe permitir:

1. Comprimir archivos de texto (.txt):

- Leer un archivo de texto plano.
- Calcular la frecuencia de aparición de cada carácter.
- Construir un árbol binario de Huffman con base en las frecuencias.
- Generar los códigos de Huffman para cada carácter.
- Codificar el contenido del archivo original con dichos códigos.
- Guardar el resultado de la compresión en dos archivos:
 - Un archivo binario .huff que contiene la codificación del archivo original.
 - Un archivo .hufftree que contiene la definición del árbol de Huffman usado para la codificación.

2. Descomprimir archivos:

- Leer los archivos .huff y .hufftree.
- Reconstruir el árbol binario de Huffman a partir del archivo .hufftree.
- Decodificar el archivo .huff utilizando el árbol reconstruido.
- Restaurar y mostrar el archivo original en formato de texto.

Requisitos Técnicos:

- Utilizar estructuras de datos (para construir el árbol de Huffman), árboles binarios, listas y mapas de frecuencias.
- El archivo .hufftree puede guardar el árbol en formato de recorrido (ej. preorden con nodos y hojas indicadas) o como una estructura serializada de tu elección.
- El archivo .huff debe ser binario y contener únicamente los datos comprimidos, sin el árbol.
- Se recomienda utilizar archivos de texto con diferentes tamaños y contenidos para las pruebas.

Tareas:

- El código fuente documentado.
- Los archivos .huff y .hufftree generados a partir de al menos dos archivos de texto diferentes.
- El archivo de texto original y su correspondiente versión descomprimida (para verificar la exactitud de la descompresión).
- Un breve informe (máximo 1 página) que explique tu implementación, decisiones de diseño y pruebas realizadas.

Debe subir a Canvas todos los productos elaborados y los enlaces a su repositorio de github (o equivalente).

Calificación:

Aspecto	Puntos
Implementación correcta de compresión	30
Implementación correcta de descompresión	30
Uso adecuado de estructuras de datos	20
Organización y claridad del código	5
Informe de entrega y archivos de prueba y pruebas unitarias	15
TOTAL:	100