

Universidad Rafael Landívar
Facultad de Ingeniería
Estructura de Datos II (01, Matutina)
Ing. Fredy Alexander Bustamante

LABORATORIO NO.3
COMPRESIÓN DE MENSAJES Y ARCHIVOS

Sergio Iván Cardona Polanco
Carné: 1222419

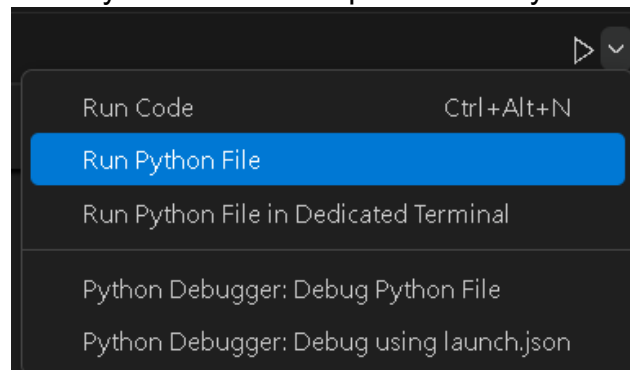
Guatemala, 30 de septiembre de 2024

DESCRIPCIÓN DEL PROGRAMA

El propósito de este programa es codificar y decodificar mensajes mediante la codificación aritmética y la codificación de Huffman, dos técnicas de compresión muy utilizadas. Los usuarios pueden codificar un mensaje original en una secuencia de bits y luego decodificarlo, lo que les permite recuperar el texto original. El programa facilita la comprensión de estos métodos de compresión en el contexto de la manipulación de datos a través de funciones para manejar la entrada y salida de datos, así como la gestión de frecuencias de caracteres.

MANUAL DE USUARIO

1. Correr el programa: Para correr el programa en Visual Studio Code, debe de dar click en la flechita y seleccionar la opción 'Run Python File'



2. A continuación, desplegará un menú para que ingrese un número y seleccione que desea realizar.

```
--- Menú Principal ---  
1. Huffman  
2. Aritmética  
3. Salir  
Selecciona una opción: 
```

3. Selección Huffman: Si seleccionó Huffman, le aparecerá el siguiente menú para que seleccione la opción que desea realizar.

```
--- Submenú Huffman ---  
1. Codificación automática  
2. Codificación no automática  
3. Decodificación  
4. Volver al menú principal  
Selecciona una opción: 
```

- 3.1.1 Si seleccionó la opción 1 (Codificación automática), le pide que ingrese el mensaje que desea codificar.

Introduce el mensaje a comprimir: Hola Mundo!

- 3.1.2 Muestra en consola resultado y luego le pide que ingrese un numero para el log del archivo

```
Mensaje codificado: 0001011111010110100111100111001011100
Tabla de frecuencias: Counter({'o': 2, 'H': 1, 'l': 1, 'a': 1, ' ': 1, 'M': 1, 'u': 1, 'n': 1, 'd': 1, 'l': 1})
Probabilidad de cada caracter: {'H': 0.09090909090909091, 'o': 0.18181818181818182, 'l': 0.09090909090909091, 'a': 0.09090909090909091, ' ': 0.09090909090909091, 'M': 0.09090909090909091, 'u': 0.09090909090909091, 'n': 0.09090909090909091, 'd': 0.09090909090909091, 'l': 0.09090909090909091}
código de cada caracter: {'H': '000', 'M': '001', 'a': '010', 'n': '011', 'd': '100', 'o': '101', 'l': '1100', ' ': '1101', 'u': '1110', 'l': '1111'}
Tasa de compresión: 0.42045454545454547
Introduce un número para el archivo de salida: 
```

- 3.1.3 Como ejemplo ingrese el numero 7 y vemos que crea el log con ese número.

Introduce un número para el archivo de salida: 7

≡ codificacion7.log U

- 3.2.1 Si seleccionó la opción 2 (Codificación no automática), le pide que ingrese el mensaje que desea codificar. Como ejemplo introduce el siguiente mensaje: A BB CCC

```
Introduce el mensaje a comprimir: A BB CCC
Introduce la tabla de frecuencias (carácter y frecuencia separados por espacio, una línea por carácter).
Nota: Para el espacio, ingresa 'espacio' seguido de su frecuencia (ejemplo: 'espacio 1').
Cuando termines, escribe 'fin'.
```

- 3.2.2 La forma correcta de introducir la tabla de frecuencias es la siguiente:

```
A 1
B 2
C 3
espacio 2
fin
```

De lo contrario de no ser ingresada correctamente o que la tabla de frecuencias este mal ingresada, le mostrará mensajes de validación de errores y pasará automáticamente a codificar el mensaje de la forma automática. Estos pueden ser algunos de los errores que se maneja:

```
A 2
B 2
C 3
espacio 2
fin
Error: La frecuencia del carácter 'A' es incorrecta. Frecuencia real: 1, pero se ingresó: 2.
Pasando a compresión automática con el mismo mensaje...
```

```
A 1
B 2
C 3
X 1
espacio 2
fin
Error: El carácter 'X' fue ingresado en la tabla de frecuencias, pero no está presente en el mensaje.
Pasando a compresión automática con el mismo mensaje...
```

```
A 1
B
Error: Debes ingresar un caracter seguido de su frecuencia. Ejemplo: 'a 3'. Inténtalo de nuevo.
B 2
C 3
espacio 2
fin
```

3.2.3 Una vez ingresada correctamente la tabla de frecuencias, le pedirá el numero para el log de la codificación no automática

```
--- Compresión No Automática ---
Mensaje codificado: 0001101001111111
Código de cada caracter: {'H': '000', 'M': '001', 'a': '010', 'n': '011', 'd': '100', 'o': '101', 'l': '1100', ' ': '01', 'u': '110', '1': '1111', 'C': '11', 'A': '00', 'B': '10'}
Introduce un número para el archivo de salida: 12
```

```
≡ codificacion12.log U
```

3.3.1 Si seleccionó la opción 3 (Decodificación), le pedirá que ingrese el nombre del log con la extensión '.log'.

```
Introduce el nombre del archivo .log: codificacion12.log
```

3.3.2. Mostrará el mensaje decodificado en la consola.

```
Mensaje decodificado: A BB CCC
```

4. Selección Aritmética: Si seleccionó Aritmética, le aparecerá el siguiente menú para que seleccione la opción que desea realizar.

```
--- Submenú Aritmética ---
1. Codificación automática
2. Codificación no automática
3. Decodificación
4. Volver al menú principal
Selecciona una opción: █
```

4.1.1 Si seleccionó la opción 1 (Codificación automática) funciona igual que la de Huffman, donde le pide el mensaje a comprimir y luego le pide el numero para el log.

- 4.2.1 Si seleccionó la opción 2 (Codificación no automática), le pide que ingrese el mensaje que desea codificar y el valor de K, donde K debe ser $\geq 4T$ (total de frecuencias). Como ejemplo introduje el siguiente mensaje: XY 1

```
Introduce el mensaje a comprimir: XY 1
Introduce la tabla de frecuencias (carácter y frecuencia separados por espacio, una línea por carácter).
Nota: Para el espacio, ingresa 'espacio' seguido de su frecuencia (ejemplo: 'espacio 1').
Cuando termines, escribe 'fin'.
```

- 4.2.2 La forma correcta de introducir la tabla de frecuencias y el valor de k es la siguiente:

```
X 1
Y 1
espacio 1
1 1
fin
Introduce el valor de k (bits por carácter): 16
```

De lo contrario de no ser ingresada correctamente o que la tabla de frecuencias este mal ingresada, le mostrará mensajes de validación de errores y pasará automáticamente a codificar el mensaje de la forma automática. Estos pueden ser algunos de los errores que se maneja:

```
Introduce el valor de k (bits por carácter): 5
Error: El valor de k debe ser al menos 4T (T = 4). k debe ser  $\geq 16$ .
Pasando a compresión automática con el mismo mensaje...
```

- 4.2.3 Una vez ingresada correctamente la tabla de frecuencias y el valor de K, le pedirá el numero para el log de la codificación no automática.

```
-- Compresión No Automática (Aritmética con Enteros) ---
Mensaje codificado (en bits): 0001101101
Tabla de frecuencias: {'X': 1, 'Y': 1, ' ': 1, '1': 1}
k (bits por carácter): 16
T (total de frecuencias): 4
n (longitud del mensaje): 4
Tasa de compresión: 0.3125
Introduce un número para el archivo de salida: 13
```

```
≡ codificacion13.log
```

```
U
```

- 4.3.1 Si seleccionó la opción 3 (Decodificación), le mostrará el mensaje decodificado en consola.

5. Si seleccionó la opción 3 en el menú principal, sale del programa.

```
--- Menú Principal ---  
1. Huffman  
2. Aritmética  
3. Salir  
Selecciona una opción: 3  
Saliendo del programa...  
PS D:\UsuarioD\Escritorio\SergioU\2semestre2024\ED2\LAB3_SergioCardona_1222419>
```

CONCLUSIONES

- La codificación de Huffman y la codificación Aritmética funcionan bien para reducir el tamaño de los datos.
- La codificación de Huffman es relativamente sencilla y eficiente en términos de tiempo de ejecución, por lo que es más fácil de implementar e interpretar.
- La codificación aritmética es más difícil de implementar debido a los cálculos y la gestión de la precisión numérica, lo que puede afectar el rendimiento en sistemas con recursos limitados.