UNIVERSIDAD DE GUADALAJARA

CENTRO UNIVERSITARIO DE CIENCIAS EXACTAS E INGENIERÍAS

DIVISIÓN DE TECNOLOGÍAS PARA LA INTEGRACIÓN CIBER-HUMANA

INGENIERÍA EN COMPUTACIÓN

Programación de Bajo Nivel D02

Entrega 1: Huffman (Front-end)

Nombre: Juan Carlos Torres Gutiérrez

Código: 222361856

Fecha: 23 de Marzo del 2024

Curso impartido por: Jorge Ernesto Lopez Arce Delgado

Algoritmo de Huffman

Horario de 07:00 a 09:00

Juan Carlos Torres Gutiérrez | Ingeniería en computación

***Abstract* — Se ha realizado una implementación de una interfaz gráfica para la utilización del algoritmo de Huffman para archivos de texto (posteriormente secuencias de bits en general)**

- Introducción

Este código implementa una interfaz gráfica de usuario (GUI) usando la biblioteca Tkinter en Python. La interfaz permite al usuario seleccionar un archivo de texto, analizar su contenido para contar la frecuencia de los diferentes caracteres (letras, dígitos, signos de puntuación, etc.) y mostrar los resultados en un widget de texto. Además, incluye funcionalidades para comprimir y descomprimir el contenido del archivo, aunque estas están marcadas como tareas pendientes.

Objetivos

El objetivo principal del código es proporcionar una herramienta simple para analizar la frecuencia de caracteres en archivos de texto seleccionados por el usuario. Los objetivos específicos incluyen:

1. Permitir al usuario seleccionar un archivo de texto.
2. Contar la frecuencia de cada carácter en el archivo seleccionado.
3. Mostrar los resultados de la frecuencia de caracteres en la interfaz gráfica.
4. Proporcionar funcionalidades para comprimir y descomprimir el contenido del archivo (aunque estas funcionalidades están incompletas y marcadas como tareas pendientes).
1.

- Desarrollo

El código comienza importando las bibliotecas necesarias: tkinter para la creación de la GUI y filedialog para la selección de archivos. También importa la biblioteca string para generar una lista de todos los caracteres posibles que se contarán en el archivo.

```
1 import tkinter as tk
2 from tkinter import filedialog
3 import string
```

A continuación, define una lista llamada characters que contiene todos los caracteres posibles que se contarán en el archivo. Luego, define una función llamada count_characters que toma un manejador de archivo como entrada, recorre el archivo línea por línea y cuenta la frecuencia de cada carácter en el archivo. Esta función devuelve un diccionario donde las claves son los caracteres y los valores son las frecuencias.

```
#List all possible characters here
characters = string.ascii_uppercase + string.ascii_lowercase + string.digits + string.punctuation

#Count the characters (nested for lmao)
def count_characters(file_handle):
    """Traverse a file and compute the number of occurences of each letter
    return results as a simple 26 element list of integers."""
    results = {char: 0 for char in characters}
    for line in file_handle:
        for char in line:
            char = char.lower()
            if char in characters:
                results[char] += 1

    return results

```

Después, define una función llamada open_file que permite al usuario seleccionar un archivo de texto, llama a la función count_characters para contar la frecuencia de caracteres en el archivo seleccionado y muestra los resultados en un widget de texto en la interfaz gráfica.

```
# Create function for later examine widget
def open_file():
    file_path = filedialog.askopenfilename(
        title="Select a Text File", filetypes=[("Text files", "*.txt")])
    if file_path:
        sourcedata = open(file_path)
        charcounts = count_characters(sourcedata)
        text_widget.delete(1.0, tk.END)
        for char, count in charcounts.items():
            content = "%s=%d\n" % (char, count)
            text_widget.insert(tk.END, content)

```

Las funciones compress y decompress serán implementadas en la segunda entrega por parte de otro compañero de la clase, que se supone que implementará algoritmos de compresión y descompresión del contenido del archivo. Estas funciones están marcadas como tareas pendientes.

```
35 # TO-DO
36 def compress():
37     text_widget.delete(1.0, tk.END)
38     content = "To do: compressing algorithm"
39     text_widget.insert(tk.END, content)
40
41
42 # TO-DO
43 def decompress():
44     text_widget.delete(1.0, tk.END)
45     content = "To do: compressing algorithm"
46     text_widget.insert(tk.END, content)
47
```

Finalmente, crea la ventana principal de la aplicación, agrega un widget de texto para mostrar los resultados, y botones para abrir un archivo, comprimir y descomprimir el contenido (aunque estas dos últimas funcionalidades no están implementadas completamente).

```
49 # Create the main window
50 root = tk.Tk()
51 root.title("Char counter")
52 # Create a Text widget to display the content
53 text_widget = tk.Text(root, wrap="word", width=40, height=10)
54 text_widget.pack(pady=10)
55 open_button = tk.Button(root, text="Examine", fg="green", command=open_file)
56 compress_button = tk.Button(root, text="Compress", fg="red", command=compress)
57 decompress_button = tk.Button(root, text="Decompress", fg="blue", command=decompress)
58 open_button.pack(pady=10)
59 compress_button.pack(pady=10)
60 decompress_button.pack()
61 root.mainloop()
```

- CONCLUSION

La dinámica de trabajar de forma activa con el programa de otra persona me parece una buena idea, el tener que adaptar las ideas de otro compañero de la clase junto con el nuestro implementando ahora la parte del algoritmo, es decir, el backend, me parece una forma de desarrollar habilidades blandas bastante buenas, espero se mantenga el ritmo de trabajo de forma retroactiva.

- Referencias

Grayson, J. E. (2000). *Python and Tkinter Programming*. https://openlibrary.org/books/OL8701912M/Python_and_Tkinter_Programming