Adrian Ruiz Arana

30/04/2024

Análisis de algoritmos

Actividad#Algoritmo de Huffman

Reporte del Programa de Compresión y Descompresión

El programa implementa una interfaz gráfica de usuario utilizando la biblioteca Tkinter de Python. Su funcionalidad principal es permitir al usuario seleccionar un archivo de su sistema, mostrar su contenido en una ventana, calcular la frecuencia de los caracteres en ese archivo y mostrar esta frecuencia en otra ventana. El objetivo de este programa es comprimir archivos de texto seleccionados por el usuario utilizando el algoritmo de codificación Hoffman y luego descomprimirlos nuevamente cuando sea necesario.

Desarrollo:

El programa está desarrollado en Python utilizando la biblioteca Tkinter para la interfaz gráfica de usuario. Se implementa el algoritmo de codificación Huffman, que consta de las siguientes etapas:

1. Cálculo de las frecuencias de los caracteres en el texto seleccionado.
2. Construcción del árbol de Huffman basado en las frecuencias calculadas.
3. Generación de códigos Huffman para cada carácter en el texto.
4. Compresión del texto original utilizando los códigos de Huffman generados.

Estructura del Código:

1. Importación de bibliotecas: Se importan las bibliotecas necesarias, como tkinter para la creación de la interfaz gráfica de usuario y heapq para la implementación de la cola de prioridad utilizada en el algoritmo de codificación Huffman.

2. Definición de la clase Nodo: Se define la clase Nodo, que representa un nodo en el árbol de Huffman. Cada nodo tiene un carácter, una frecuencia y enlaces a sus hijos izquierdo y derecho.
3. Funciones de cálculo de frecuencias y construcción del árbol de Huffman: Se definen funciones para calcular las frecuencias de los caracteres en un texto y para construir el árbol de Huffman a partir de estas frecuencias. El árbol se construye utilizando una cola de prioridad (heap) donde los nodos con menor frecuencia tienen mayor prioridad.
4. Función de generación de códigos de Huffman: Se define una función para generar los códigos de Huffman para cada carácter en el árbol de Huffman. Estos códigos se utilizan para comprimir el texto original.
5. Función de compresión: Se define una función para comprimir el texto original utilizando los códigos de Huffman generados.
6. Función de descompresión: Se define una función para descomprimir el texto comprimido utilizando los códigos de Huffman. Esta función invierte el diccionario de códigos para encontrar el carácter correspondiente a cada código y reconstruye el texto descomprimido.
7. Funciones de la interfaz gráfica de usuario (GUI): Se definen funciones para interactuar con el usuario a través de la interfaz gráfica. Estas funciones permiten al usuario seleccionar archivos, comprimir y descomprimir texto, y mostrar los resultados en la GUI.
8. Configuración de la GUI y bucle principal: Se configura la interfaz gráfica de usuario con widgets como botones y cuadros de texto, y se inicia el bucle principal de la aplicación que mantiene la ventana abierta y receptiva a las interacciones del usuario.

Definición de Funciones:

examinar_archivo(): Esta función se ejecuta cuando se hace clic en el botón "Examinar". Utiliza la función askopenfilename de filedialog para permitir al usuario seleccionar un archivo. Luego, lee el contenido del archivo seleccionado y lo muestra en un área de texto. Posteriormente, llama a la función calcular_frecuencia para calcular la frecuencia de los caracteres en el archivo.

1. calcular_frecuencias(texto):

Esta función calcula la frecuencia de cada carácter en el texto dado dado un archivo. texto (str) - El texto del cual se calcularán las frecuencias donde Un diccionario donde las claves son los caracteres y los valores son las frecuencias de esos caracteres en el texto.

2. construir_arbol(frecuencias):

Esta función construye el árbol de Huffman a partir de las frecuencias de los caracteres. frecuencias (dict) - Un diccionario que contiene las frecuencias de los caracteres en el texto.

3. generar_codigos(nodo, prefijo="", codigo={}):

Esta función genera los códigos de Huffman para cada carácter en el árbol de Huffman.

nodo (Nodo) - El nodo actual en el árbol de Huffman.

prefijo (str) - El prefijo actual para el código.

codigo (dict) - Un diccionario para almacenar los códigos generados.

4. comprimir(texto, codigos):

Esta función comprime el texto original utilizando los códigos de Huffman generados.

- Entrada:
  - texto (str) - El texto original que se va a comprimir.
  - codigos (dict) - Un diccionario que contiene los códigos de Huffman para cada carácter.
- Salida: Una cadena de bits que representa el texto comprimido.

5. descomprimir(texto_comprimido, codigos):
   - Descripción: Esta función descomprime el texto comprimido utilizando los códigos de Huffman.
   - Entrada:
     - texto_comprimido (str) - El texto comprimido que se va a descomprimir.
     - codigos (dict) - Un diccionario que contiene los códigos de Huffman para cada carácter.
   - Salida: El texto descomprimido.
6. abrir_archivo():
   - Descripción: Esta función se ejecuta al presionar el botón "Examinar y Comprimir" en la interfaz gráfica de usuario. Permite al usuario seleccionar un archivo de texto, calcular las frecuencias de los caracteres, construir el árbol de Huffman, generar los códigos de Huffman, comprimir el texto y guardar el texto comprimido en un archivo binario.
7. descomprimir_archivo():

- Descripción: Esta función se ejecuta al presionar el botón "Descomprimir" en la interfaz gráfica de usuario. Permite al usuario seleccionar un archivo binario comprimido, descomprimirlo utilizando los códigos de Huffman y guardar el texto descomprimido en un archivo de texto**.**

Configuración de la Interfaz Gráfica:

Se crea una instancia de Tk y se establece el título de la ventana.

Se crea un botón llamado "Examinar" que llama a la función examinar_archivo.

Se crean dos áreas de texto: una para mostrar el contenido del archivo seleccionado y otra para mostrar las frecuencias de los caracteres.

Ejecución del Programa:

Se inicia el bucle principal de Tkinter con mainloop().

Funcionamiento:

El usuario hace clic en el botón "Examinar" para seleccionar un archivo de su sistema.

El programa lee el contenido del archivo seleccionado y lo muestra en el primer área de texto.

Calcula la frecuencia de los caracteres en el contenido y muestra estos resultados en el segundo área de texto.

Los resultados también se guardan en un archivo llamado "frecuencias.txt" en el mismo directorio donde se ejecuta el programa.

Conclusión:

El programa proporciona una forma simple y eficiente de examinar archivos, visualizar su contenido y calcular la frecuencia de los caracteres presentes en ellos. La interfaz gráfica facilita su uso para usuarios con poca experiencia técnica, mientras que el proceso de cálculo de frecuencia proporciona una funcionalidad útil para el análisis de datos en archivos de texto.