Informe del Proyecto “Desafío 1”

Kevin Esteban Echeverri Carmona, 1000294820
Ivonne Lizeth Rosero Cardona, 1007687589
(Universidad de Antioquia)

I. ANALISIS DEL PROBLEMA Y CONSIDERACIONES PARA ALTERNATIVA DE SOLUCION.

El problema central consiste en **recuperar archivos encriptados y comprimidos** mediante dos métodos distintos: **RLE** y **LZ78**. El reto principal es que ni la clave de encriptación ni la rotación de bits utilizada son conocidas, por lo cual el algoritmo debe **probar todas las combinaciones posibles**.

Una vez logrado el desencriptado, el tratamiento difiere según el esquema de compresión utilizado:

RLE: tras desencriptar, se genera una **pista comprimida del texto original** para compararla con la salida y validar coincidencias.

LZ78: se requiere **desencriptar y descomprimir completamente**, ya que la reconstrucción depende de un **diccionario dinámico con contexto**. Solo después se puede comparar la salida con la pista original.

II. ESQUEMA DE TAREAS EN EL DESARROLLO DE LOS ALGORITMOS

A. ENTRADA DE DATOS

Solicitar al usuario la cantidad de archivos y cargar cada archivo a la memoria luego para ser llamado por una función de lectura de archivos.

B. ENCRIPTACION Y DESENCRIPTACION

En este algoritmo se hace primero probar todas las rotaciones y así mismo para cada rotación se prueba una clave, para generar una posible coincidencia.

C. FLUJO SEGÚN EL METODO DE COMPRESION

Con RLE el flujo es comprimir y encriptar la pista propuesta en la guía de desarrollo y comparar con el texto encriptado.

El flujo en LZ78 fue distinto y es que se desencripta, una vez desencriptado el texto se valida que la estructura del arreglo corresponda a la estructura planteada, donde se conoce que se tiene un separador, un índice y un carácter, una vez validada esta estructura se procedió a descomprimir, y finalmente se comparó el resultado del texto desencriptado y descomprimido con nuestra pista original.

D. VERIFICACION

Comprobar con cada una de las combinaciones de clave y rotación si produce una coincidencia valida.

E. SALIDA DE DATOS

Cuando se llega a una coincidencia entre la pista y el encriptado, se visualiza en la terminal el texto desencriptado y descomprimido, el método de compresión utilizado y la clave y rotación de bits utilizada para la encriptación.

III. ALGORITMOS IMPLEMENTADOS

A. ALGORITMO DE DESENCRIPTACION

- Entrada: archivo, clave candidata, rotación de bits.

- Operación: XOR + rotación de bits.

- Salida: texto en estado comprimido.

B. ALGORITMO DE VALIDACIÓN PARA RLE

Se lee la pista comprimida, la cual tiene una estructura de separador, número y carácter, donde el separador es el carácter 0x00. Luego procede a encriptar la entrada con una rotación n y una clave de 0xXX, y posterior a esto se procede a comparar esta pista encriptada con el texto encriptado, así sucesivamente con n de 1 a 7 y 0xXX de 0x00 a 0xFF.

C. ALGORITMO DE DESCOMPRESIÓN Y VALIDACIÓN PARA LZ78

Se realizan pruebas con diferentes claves y rotaciones hasta lograr desencriptar el texto de manera que adopte el formato correcto (separador–índice–carácter). Una vez validado dicho formato, se procede a efectuar la descompresión y posteriormente se compara la pista con el texto desencriptado y descomprimido.

D. ALGORITMO DE LECTURA DE ARCHIVOS

Se recibe la ruta y el nombre del archivo a leer. Posteriormente, el contenido es almacenado carácter por carácter, asignando un byte a cada uno. Finalmente, la función retorna la dirección del primer carácter junto con el tamaño total del arreglo.

IV. PROBLEMAS DE DESARROLLO AFRONTADOS

Uno de los principales problemas identificados durante el desarrollo de la solución fue que la comparación entre la pista y el texto original comprimidos no coincidía de manera exacta.

Adicionalmente, la correcta asignación de los tipos de variables representó un reto, ya que al trabajar con operaciones a nivel binario era necesario considerar las restricciones impuestas por el rango de cada tipo de dato.

En cuanto a la lectura de archivos, se presentó la dificultad de trabajar con el estándar Windows-1252. Este estándar gestiona el ASCII extendido de forma tal que cada carácter ocupa un byte, a diferencia del estándar UTF-8, donde la cantidad de bytes por carácter puede variar.

V. EVOLUCION DE LA SOLUCION Y CONSIDERACIONES PARA LA IMPLEMENTACION

Durante la implementación fue necesario establecer criterios específicos de validación, particularmente en los procesos de compresión mediante los algoritmos RLE y LZ78. En la resolución de los problemas identificados se privilegió en todo momento la búsqueda de la solución óptima. Para el caso del algoritmo RLE, se consideró adecuado comprimir y encriptar la pista con el fin de compararla posteriormente con el texto original comprimido y encriptado, centrando la validación en la correspondencia entre el separador y el carácter. Esta decisión metodológica se fundamenta en que, si bien el algoritmo RLE puede generar variaciones en el índice de repetición, mantiene inalterado el patrón de los caracteres, lo que permite una comparación confiable.

En el caso del algoritmo LZ78 su solución es que se trabajara con textos planos para poder encontrar coincidencias, desencriptar y descomprimir mi texto original con las distintas claves y rotaciones para llegar a encontrar similitud con la pista.