

Desafío I – Descompresión y Descriptación

Cristian Florez Monsalve

Thomas Mejia Moncada

Grupo 6

Informática II

Aníbal Jose Guerra Soler

Universidad de Antioquia

Colombia-Medellín

18/09/2025

Análisis del Problema

Un mensaje encriptado es entregado en forma de un archivo de texto plano, mensaje producto de una serie de pasos de compresión y desplazamiento de bits, el mensaje encriptado y comprimido se compone de letras alfabéticas mayúsculas, minúsculas y de números de cero al nueve. El mensaje primero fue comprimido usando alguno de estos dos métodos LRE o LZ78 posterior a comprimir el mensaje se encripto aplicando una máscara de un byte desconocida denominada como K, posterior a eso se camufló aun mas el mensaje usando desplazamiento de bits a la *Izquierda* una cantidad de n veces. Adicional recibimos otro documento de texto plano que incluye un fragmento de lo que fue el texto original.

El objetivo es aplicar de forma inversa la descryptación de forma tal que se logre conocer el valor de los n bits que se desplazaron a la izquierda, encontrar la llave k que fue usada a la hora de enmascarar el mensaje, y descubrir cuál de los métodos de compresión fue utilizado a la hora de comprimir el mensaje ya sea LRE o LZ78, una vez hechos todos los pasos se debe verificar que el fragmento entregado como pista corresponda efectivamente a una fracción del texto descryptado y descomprimido.

información de entrada:

- **Archivo .txt:** Contiene el mensaje encriptado y comprimido.
- **Archivo .txt:** Contiene el fragmento del mensaje original.

Información de salida:

- **Archivo .txt:** Contiene el mensaje original después de descryptarlo.

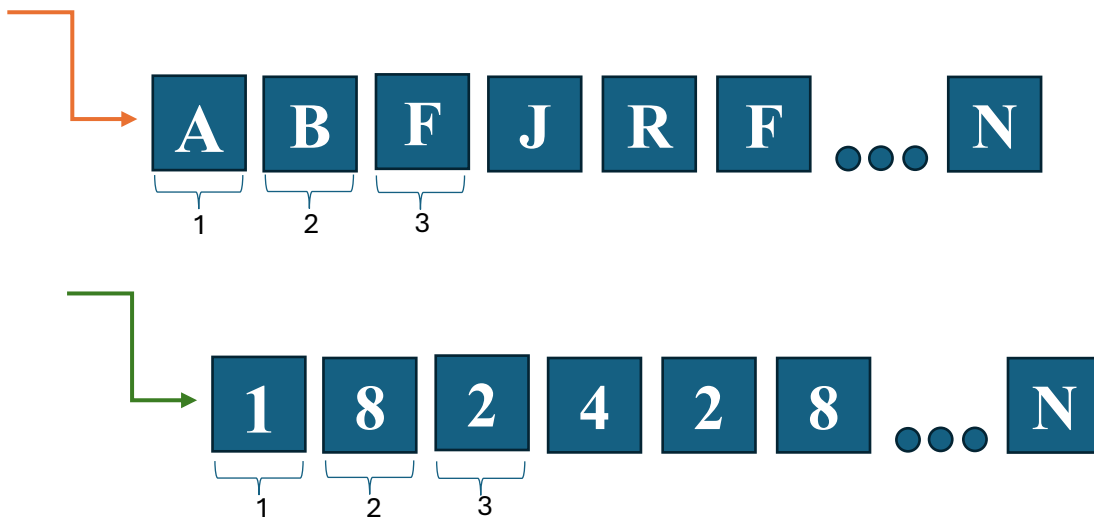
Consideraciones a tener en cuenta:

1. Se deben usar 2 bytes para almacenar de forma correcta y evitar desbordes de bits a la hora de descomprimir los prefijos.
2. Debe incluir un uso adecuado y consciente de la memoria dinámica y el uso de estructuras de datos complejas o simplificadas según lo requiera la solución.

Primer Propuesta: Desarrollo y Solución

Utilizamos un algoritmo de fuerza bruta para buscar un resultado funcional, aunque costoso en el uso de recursos por el alto costo en procesamiento lo que se traduce en mayor tiempo de ejecución.

Estructura de datos: Usaremos 2 arreglos unidimensionales dinámicos desacoplados para hacer un uso sencillo y rápido de los índices y prefijos del texto encriptado, de tal forma que un arreglo corresponde a los datos de tipo carácter que hacen parte del texto, y un arreglo a datos del tipo entero que corresponderán a los prefijos que acompañan los caracteres. Ambos arreglos, aunque desacoplados siguen un orden de índice, donde el elemento del n-esimo índice del primer arreglo corresponde con el elemento del n-esimo índice del segundo arreglo.



Para llevar acabo la solución del problema seguiremos un algoritmo simple que actúa de forma secuencial, así:

1. Leemos el archivo donde se encuentra el texto encriptado y comprimido.
2. Cargamos la información en la estructura de datos.
3. Usamos un ciclo para iterar sobre cada componente que hace parte de la estructura de datos.
4. Luego usamos otro ciclo para iterar desde 0 a 255 de forma binaria para probar todas las posibles keys K que pudieron haber sido usadas.
5. Usamos XOR sobre la key obtenida y cada elemento del arreglo de caracteres.
6. Aplicamos un desplazamiento a la izquierda o a la derecha según nos sea mas conveniente.
7. Probamos con ambos métodos de descompresión LRE o LZ78
8. Iteramos sobre el texto resultante buscando una coincidencia exacta con la pista (fragmento del texto original) si lo encontramos entonces tomamos los valores de k, n y el método de descompresión que usamos

9. Retornamos al usuario tanto los valores usados para solucionar y el texto original después de descomprimir y descriptar
10. Si no coincide se repite el proceso hasta hallar un k , n y el método de descompresión que fue usado.

Diagrama de flujo

