

Proyecto Entrega 2: Plan de Pruebas

Guillermo Andrés Aponte Cárdenas

Daniel Felipe Ramírez Vargas

Pontificia Universidad Javeriana

Estructuras de Datos

Bogotá, 2025

Comando “decodificar”

Descripción

El comando debe cargar en memoria las secuencias contenidas en el archivo binario `nombre_archivo.fabin` , que contiene una codificación Huffman de un conjunto de secuencias en el formato descrito más arriba. Si dentro de la misma sesión de trabajo ya se han cargado otros archivos de secuencias (usando el comando `cargar`), la información debe sobrescribirse en memoria, es decir, no se deben combinar informaciones de secuencias de diferentes archivos. Si por alguna razón no es posible cargar la información de codificación, o no es posible realizar el proceso de decodificación, el comando debe mostrar el mensaje de error.

Casos de Prueba

Descripción Caso	Valores de Entrada	Resultado Esperado	Resultado Obtenido
Caso normal, decodificar un archivo existente.	Nombre del archivo = archivo con la información necesaria	Mensaje: Secuencias decodificadas desde <code>nombre_archivo.fabin</code> y cargadas en memoria	SECUENCIA DECODIFICADA EXITOSAMENTE DESDE EL ARCHIVO “ARCHIVO.FABIN”

Caso vacío, decodificar un archivo inexistente	Nombre del archivo = archivo inexistente	Mensaje: No se pueden cargar las secuencias desde nombre_archivo.fabin .	No se pueden cargar ls secuencias desde archivoinexistente.fabin
Caso erróneo, decodificar un archivo corrupto	Nombre del archivo = archivo corrupto	Mensaje: mensaje de error desde la consola	Violación de segmento (core dumped) [Núcleo vaciado a un archivo]

Evidencias de Ejecución

Descripción de caso	Evidencia de ejecución
Caso normal, decodificar un archivo existente..	<pre> \$decodificar archivo.fabin Letra: C codificacion: 011 Letra: T codificacion: 100 Letra: G codificacion: 0000 Letra: A codificacion: 111 Letra: U codificacion: 0100 Letra: R codificacion: 00010 Letra: D codificacion: 0011 Letra: N codificacion: 1011 Letra: S codificacion: 00011 Letra: H codificacion: 00101 Letra: V codificacion: 1010 Letra: K codificacion: 1101 Letra: M codificacion: 11000 Letra: B codificacion: 0101 Letra: Y codificacion: 11001 Letra: W codificacion: 00100 GUARDANDO SECUENCIA DECODIFICADA EN MEMORIA... SECUENCIA DECODIFICADA EXITOSAMENTE DESDE EL ARCHIVO " archivo.fabin " \$ </pre>
Caso vacío, decodificar un archivo inexistente	<pre> \$decodificar archivoinexistente.fabin No se pueden cargar las secuencias desde archivoinexistente.fabin \$ </pre>

<p>Caso</p> <p>erróneo,</p> <p>decodifica</p> <p>r un</p> <p>archivo</p> <p>corrupto</p>	<pre>\$decodificar archivo_corrupto.fabin Letra: C codificacion: 011 Letra: T codificacion: 100 Letra: G codificacion: 0000 Letra: A codificacion: 111 Letra: U codificacion: 0100 Letra: R codificacion: 00010 Letra: D codificacion: 0011 Letra: N codificacion: 1011 Letra: S codificacion: 00011 Letra: H codificacion: 00101 Letra: V codificacion: 1010 Letra: K codificacion: 1101 Letra: M codificacion: 11000 Letra: B codificacion: 0101 Letra: Y codificacion: 11001 Letra: W codificacion: 00100 make: *** [Makefile:7: CompilaryEjecutar] Violación de segmento (core dumped) [Núcleo vaciad</pre>
--	---