

Universidad Rafael Landívar

Facultad de Ingeniería

Estructuras de datos II

Sección: 03

Catedrático: Ing. Boris José Búcaro Pazzetti

Lab1:

Codificación Huffman y Compresión Aritmética

Dereck Alexander Cabrera Ng – 1177223

24 de septiembre de 2024

Resolución de preguntas

- a. ¿Qué algoritmo es mejor para comprimir estos datos y por qué?
Si consideramos únicamente el tamaño final del archivo, la compresión de Huffman fue la más eficiente, sin embargo, si hablamos respecto al tiempo que necesitan ambos algoritmos Huffman termina siendo más lento. Lo importante en este caso es el tamaño final pero hay que considerar el tiempo que puede llegar a tomar.
- b. Si el mejor algoritmo para comprimir fue A que se debería de considerar para que B fuera mejor algoritmo, ¿qué cambios se deberían hacer en el algoritmo?
Para que la compresión aritmética supere a la de Huffman, es fundamental considerar la entropía de los datos. A medida que el tamaño del conjunto de datos aumenta, la compresión aritmética se ajusta más a la entropía teórica mínima, lo que le permite generar una compresión más eficiente. Esto se debe a que la compresión aritmética puede manejar probabilidades fraccionarias con mayor precisión, lo que la hace más efectiva en conjuntos de datos grandes o con distribuciones de símbolos más uniformes.
- c. ¿Cómo se vio afectado el programa del Lab01 con estos nuevos requerimientos, vale la pena la compresión de datos?
Hubo dos cambios esenciales al Lab01, el primer cambio fue adaptar la clase que guarda los libros para guardar el valor de la compresión, el segundo fue el escribir al archivo cual fue el mejor algoritmo. Si lo que se busca es reducir el tamaño necesario para guardar los libros, entonces si vale la pena comprimir los datos.
- d. ¿Qué recomendaciones haría para mejorar los algoritmos vistos en clase?
Se puede implementar una rutina de análisis previo que evalúe la distribución de frecuencias de los datos antes de seleccionar qué algoritmo utilizar. Esto permitiría elegir automáticamente entre Huffman y la compresión aritmética, optimizando así tanto el tamaño comprimido como el tiempo de ejecución según las características de los datos.
Otra opción es la de utilizar multi-threading para así reducir el tiempo de ejecución.