Universidade Federal do Rio Grande do Norte

Centro de Ciências Exatas e da Terra

Bacharelado em Ciência da Computação

DIM0117 - Estruturas de Dados Básicas II

Relatório Algoritmo de Compressão

Autores: Luisa Ferreira de Souza Santos, Yuri Maximiliano Brasileiro Santos

Professor: André Maurício Cunha Campos

Natal – RN

Outubro de 2025

1 Análise de complexidade

Encoding (Compressão)

- **Inserção dos símbolos na árvore:** $O(n)$, sendo n a quantidade de caracteres no arquivo.
- **Leitura dos bytes do arquivo:** $O(n)$, sendo n o número total de bytes lidos.
- **Contagem de frequência dos símbolos:** $O(n)$, sendo n a quantidade total de caracteres do texto.
- **Construção da árvore de Huffman:** $O(n \log n)$, sendo n o número de símbolos distintos. Essa etapa envolve a criação da fila de prioridade e a combinação iterativa dos nós.
- **Geração da tabela de códigos:** $O(n)$, sendo n o número de símbolos distintos. A tabela é obtida percorrendo a árvore e associando a cada símbolo o seu respectivo código binário.
- **Escrita dos bits codificados:** $O(n)$, pois o algoritmo percorre o texto uma única vez, convertendo cada símbolo em sua representação binária. Como o tamanho médio dos códigos de Huffman (L) é constante, o custo total é proporcional a n.

Decoding (Descompressão)

- **Leitura da árvore de Huffman:** $(|\text{símbolo}| \times 8 + 8) \times n = O(n)$. A reconstrução da árvore ocorre a partir da leitura da sua codificação binária.
- **Leitura e escrita dos símbolos:** $O(n)$, pois o algoritmo lê os bits um a um, percorre a árvore até alcançar uma folha e escreve o símbolo correspondente. Como o comprimento médio dos códigos (L) é constante, a complexidade permanece aproximadamente $O(n)$.

2 Comparação com outros compressores

A taxa de compressão (%) é calculada como:

$$\text{Taxa de Compressão (\%)} = 1 - \frac{\text{Tamanho Comprimido}}{\text{Tamanho Original}} \times 100$$

2.1 Arquivos de Texto

2.1.1 Texto curto (3.965 caracteres / 3.189 bytes)

Método	Tamanho (bytes)	Taxa de Compressão (%)
Huffman	1.951	38.8
ZIP	1.606	49.6
GZIP	1.456	54.3
7z	1.613	49.4

2.1.2 Texto grande (9.855 caracteres / 8.328 bytes)

Método	Tamanho (bytes)	Taxa de Compressão (%)
Huffman	4.970	40.3
ZIP	3.413	59.0
GZIP	3.272	60.7
7z	3.351	59.8

2.2 Arquivos C++

2.2.1 Arquivo com 9.165 bytes

Método	Tamanho (bytes)	Taxa de Compressão (%)
Huffman	5.869	36.0
ZIP	2.754	69.9
7z	2.655	71.0
GZIP	2.614	71.5

2.2.2 Arquivo com 11.541 bytes

Método	Tamanho (bytes)	Taxa de Compressão (%)
Huffman	6.994	39.4
ZIP	3.459	70.0
GZIP	3.312	71.3
7z	3.338	71.1

2.3 Arquivo Python com 6.575 bytes

Método	Tamanho (bytes)	Taxa de Compressão (%)
Huffman	3.957	39.8
ZIP	2.094	68.2
GZIP	1.954	70.3
7z	2.066	68.6

2.4 Arquivo Java com 3.527 bytes

Método	Tamanho (bytes)	Taxa de Compressão (%)
Huffman	2.002	43.2
ZIP	934	73.5
GZIP	797	77.4
7z	919	73.9

3 Conclusões

Apesar de funcional, o compressor de Huffman apresenta, em média, uma taxa de compressão de cerca de 40%. Apesar disso, sua eficiência é limitada quando comparada à outras compressores, como, por exemplo, ZIP, GZIP e 7z, que possuem padrões de repetições mais complexos. Além

disso, o algoritmo possui complexidade de compressão de $O(...)$ e de descompressão de $O(..)$, sendo, portanto, bastante eficiente em termos computacionais.