

Relatório do projeto da disciplina de Introdução à Teoria da Informação Implementação do algoritmo de compressão e descompressão Lempel-Ziv-Welch (LZW) utilizando a linguagem de programação C++

Nathan Rodrigues Tavares de Lima
Universidade Federal da Paraíba
Depto. de Sistemas para Computação
Engenharia da Computação
João Pessoa, Paraíba
Email: nathanlima@eng.ci.ufpb.br

Wesley Dezidério da Silva
Universidade Federal da Paraíba
Depto. de Sistemas para Computação
Engenharia da Computação
João Pessoa, Paraíba
Email: wesleydeziderio@eng.ci.ufpb.br

Abstract—This report seeks to present and discuss the results obtained during the development of the Lempel-Ziv-Welch compression and decompression algorithm (LZW) in the discipline of Introduction to Information Theory taught by Professor Derzu Omaia. Using the C++ programming language, we will elaborate on the main topics specified during this development.

November 17, 2022

Resumo—Este relatório busca apresentar e discutir os resultados obtidos durante o desenvolvimento do algoritmo de compressão e descompressão Lempel-Ziv-Welch (LZW) na disciplina de Introdução à Teoria da Informação ministrada pelo professor Derzu Omaia. Utilizando a linguagem de programação C++, iremos elaborar os principais tópicos especificados durante esse desenvolvimento.

17 de novembro, 2022

1. Introdução

Para desenvolver o projeto, utilizamos a Linguagem de programação C++, haja vista que em um momento inicial chegamos à conclusão de que seria melhor trabalharmos com uma linguagem que apesar de não ser de fácil manipulação comparada a outras linguagens (como Python, por exemplo) achamos válida sua escolha. Todas as funcionalidades apresentadas foram desenvolvidas manualmente, utilizando apenas as bibliotecas padrão do C++.

2. Compressão e descompressão

No nosso cenário, realizamos o teste de compressão e descompressão com dois arquivos distintos: um arquivo de texto e um arquivo de vídeo do tipo MP4.

2.1. Compressão do arquivo de texto

Inicialmente, obtivemos problemas com o arquivo de texto em que ao descomprimir os arquivos não eram idênticos, pois algumas letras não estavam aparecendo no arquivo decodificado. Isso se dava ao fato da leitura das linhas não estar sendo feita de maneira correta. Após alterações no modo em que as linhas do arquivo estavam sendo lidas, o erro deixou de ocorrer. Além disso, utilizado o recurso de linha de comando de sistemas UNIX, diff, foi possível constatar que o arquivo original e o arquivo decodificado eram idênticos.

2.2. Compressão do arquivo de vídeo

Já para o arquivo de vídeo tivemos o seguinte problema de não conseguirmos salvar o primeiro bit. Consequentemente durante a decodificação, esse bit fazia com que apesar de o arquivo ser decodificado para o tamanho suficientemente próximo ao original, ele não era passível de ser reproduzido, o que é uma característica indesejável do compressor. Após uma tratativa semelhante a do arquivo de texto, esse bit passou a ser codificado e decodificado normalmente e o arquivo de vídeo passou a ser reproduzível.

3. Aplicação dos valores de 'K'

Sim. Todos os valores de K foram possíveis de serem aplicados, tanto para o arquivo de texto quanto para o arquivo de vídeo.

4. Quantidade exata de bits

Nesse aspecto, obtivemos êxito ao salvar a quantidade exata de bits utilizando a técnica de conversão de bits para strings.

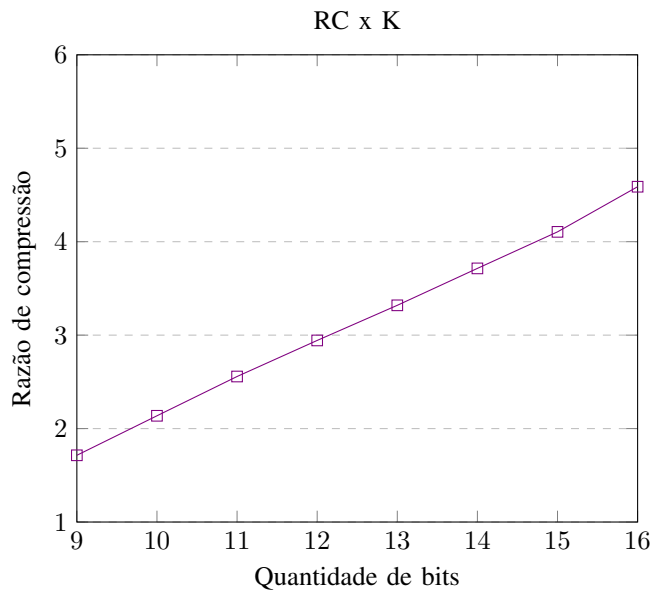
5. Curvas de RC x K

Calculamos a razão de compressão de todos os arquivos baseado na seguinte expressão:

$$RC = \frac{\text{tamanhoArquivoOriginal}}{\text{tamanhoArquivoComprimido}}$$

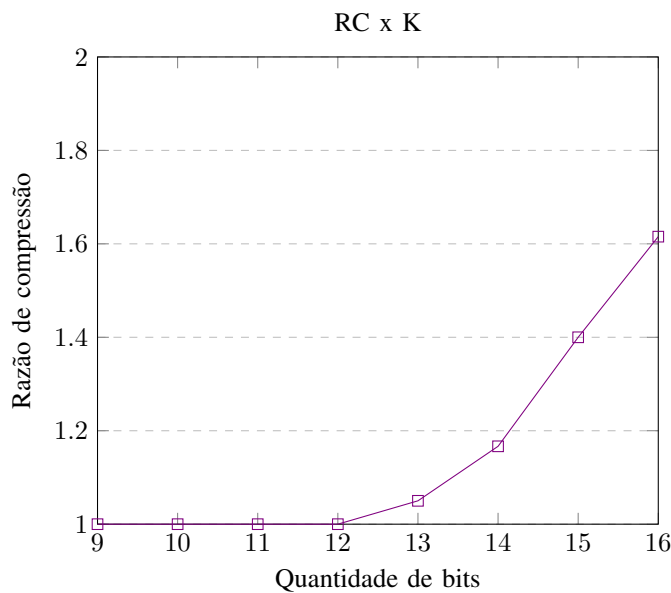
5.1. Curva de RC x K para o arquivo de texto

e obtivemos o seguinte gráfico:



5.2. Curva de RC x K para o arquivo de vídeo

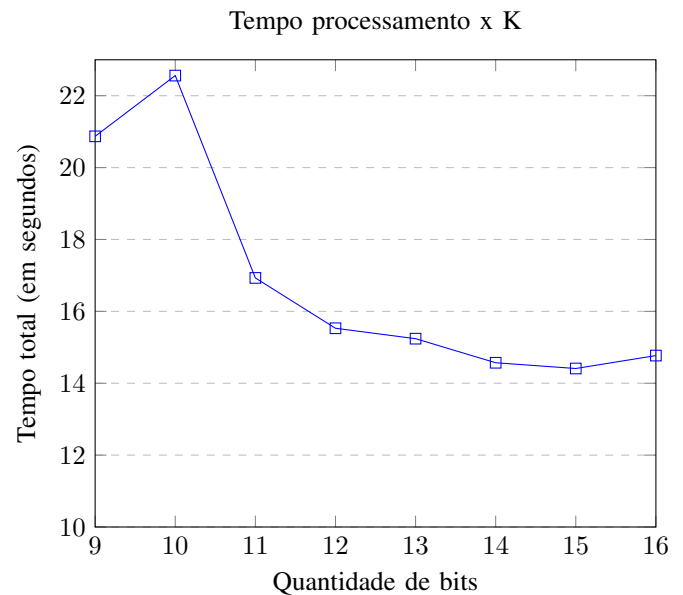
Para esse caso, obtivemos o seguinte gráfico.



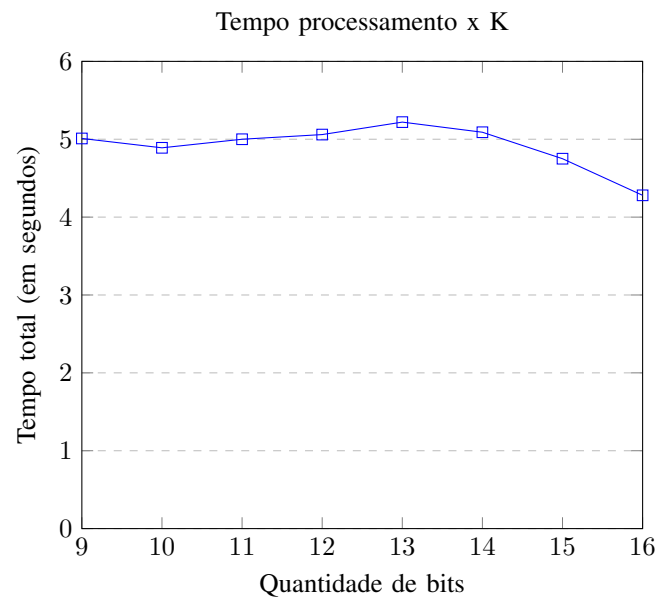
6. Curvas de Tempo x K

Para a curva de tempo x K, realizamos o somatório de todo o tempo de processamento.

6.1. Curva de Tempo x K para o arquivo de texto



6.2. Curva de Tempo x K para o arquivo de vídeo



7. Estaticidade do dicionário

Depois que o dicionário atinge o seu tamanho máximo, ele permanece estático. Salientando que seu valor máximo depende do 2^k , temos como valor máximo $2^{16} = 65536$

Referências

- [1] MIT. MIT 6.02 DRAFT Lecture Notes, Feb 2012. URL: <https://web.mit.edu/6.02/www/s2012/handouts/3.pdf>.
- [2] Amartya Ranjan Saikia. Lzw Compression Technique, Nov 2022. URL: <https://www.geeksforgeeks.org/lzw-lempel-ziv-welch-compression-technique>.