Trabalho Prático 1 – Algoritmos 2 - DCC207

Marco Túlio de Pinho Tavares Tristão

Departamento de Ciência da Computação - Universidade Federal de Minas Gerais (UFMG)
Belo Horizonte – MG – Brasil

marcotuliopin@ufmg.br

Nesse trabalho prático, implementamos uma versão do algoritmo LZ78 usando a estrutura de dados Trie. O trabalho em questão foi criado em Python 3, usando o sistema operacional Linux Ubuntu.

Implementação da Compressão

Para a compressão de arquivo de texto, usamos um dicionário (implementado com uma Trie) para armazenar as substrings já vistas no texto e associamos um código único a cada substring.

A leitura do arquivo é feita da seguinte forma: percorremos o arquivo de entrada, caractere por caractere, e checamos se essa substring está no dicionário. Caso esteja, adicionamos o próximo caractere do arquivo ao fim da substring e repetimos o processo até encontrarmos uma substring nova. Ao encontrarmos uma substring nova, a adicionamos ao dicionário. Para cada substring adicionada, acrescentamos um par (código, substring) a um vetor v.

Ao fim da leitura, convertemos cada par para suas representações em binário e escrevemos essas representações no arquivo de saída. As representações binárias terão o comprimento, em bytes, do número de bytes necessários para representar o maior código, no caso dos códigos, e do número de bytes necessários para representar o caractere com maior código UTF-8, no caso dos caracteres. As informações do tamanho dos códigos e do tamanho dos caracteres serão as primeiras informações escritas no arquivo de saída.

Implementação da Descompressão

Inicialmente, iniciamos um dicionário (implementado com uma Trie), tendo somente o nó raiz, que possui código 0 e a substring vazia.

Ao recebermos um arquivo compactado, lemos o tamanho dos códigos e o tamanho dos caracteres. Após isso, lemos um código e o convertemos para um inteiro e lemos um caractere e o convertemos para um char (o arquivo compactado estará escrito na ordem código-caractere). Com essas informações, buscamos a palavra correspondente ao código no dicionário e acrescentamos o caractere lido ao fim dela, escrevendo o resultado no arquivo de saída. A nova substring também será adicionada ao dicionário,

e o seu código será o número de palavras no dicionário mais um. Repetimos esse processo até o fim da leitura do arquivo compactado.

Taxas de Compressão

constituicao1988.txt: 33,06%

dom_casmurro1988.txt: 27,85%

dracula.txt: 18,20%

geshukunin.txt: 60,74%

great_gatsby.txt: 27,24%

greek.txt: 19,82%

metamorphosis.txt: 22,26%

moby_dick.txt: 23,07%

os_lusiadas.txt: 44,03%

romeo_and_juliet.txt: 19,86%

the_prince.txt: 27,49%