

Documentação TP1

Manipulação de sequências

1.Introdução

O trabalho consiste na implementação de um algoritmo que comprime textos em txt para lz78 e também os descomprime de volta para txt. A compressão é baseada no algoritmo LZ78 e implementada usando uma árvore trie para organizar os termos no dicionário.

2.Implementação

Primeiramente, são definidas uma struct, para os nós da árvore e uma classe Trie. A classe Trie possui duas funções, a insert, que insere novos nós, e a getIndex que retorna o índice associado ao nó que contém o termo buscado.

Em seguida, são implementadas duas funções, a compress e a decompress que comprimem e descomprimem a entrada respectivamente. Ambas as funções são divididas em três seções, uma para a leitura do arquivo de entrada, outra para a compressão, ou descompressão do arquivo e a terceira para a escrita no arquivo de saída.

O código implementado consegue transformar textos em ascii e unicode para a forma requerida do algoritmo LZ78, imprimindo pares (index, char), mas na forma index^char para uma maior compressão. Em seguida, o código consegue descomprimir e retornar o texto original exatamente. Infelizmente, ainda assim, não consegui fazer com que o código comprimido tivesse tamanho menor que o original em todos os casos. Segue abaixo a taxa de compressão de cada exemplo:

Lusíadas 348kb -> 403kb

Constituição 643kb ->570kb

Dom Casmurro 409kb -> 464kb

Divina Comédia 642kb -> 715kb

Moby Dick 1.243 ->1.369