

# TP1 - Compressão de Arquivos

## Identificação

**Nome:** Bárbara Martins Ribeiro Duarte

**Matrícula:** 2021074077

## Implementação

O algoritmo original (LZ78) utiliza de um dicionário para diminuir a quantidade de caracteres necessários no arquivo, reutilizando os caracteres que já apareceram previamente no texto para acessá-los e associá-los a novos sufixos, diminuindo o tamanho do arquivo.

Na nossa implementação, no lugar de um dicionário, utilizamos a Trie. A árvore é uma forma mais prática de buscar os caracteres e prefixos que já apareceram no texto e adicionamos a ela os novos sufixos. Ou seja, geramos um array de tuplas que contém o tamanho em bytes de um índice a ser acessado, o índice que indica o prefixo e o caractere do sufixo.

Assim, para descomprimir um arquivo, é necessário apenas acessar primeiro o índice do prefixo de forma recursiva e imprimí-lo até que seja possível imprimir o sufixo, fazendo assim para todo o array de tuplas que temos armazenado.

## Casos de Teste

Dicionario: 1067 KB => 676KB - 36.6%

Frankenstein: 429KB => 300 KB - 30.0%

JaneEyre: 934KB => 645KB - 30.9%

LoremIpsum2MB: 2000KB => 691KB - 65.4%

MarquezDePombal: 77KB => 66KB - 14.2%

MobyDick: 1232KB => 842KB - 31.6%

NotreDameDeParis: 1070KB => 718KB - 32.9%

OsLusiadas: 337KB => 248KB - 26.4%

PrideAndPrejudice: 758KB => 469KB - 38.1%

ThePictureOfDorianGrey: 443KB => 309KB - 30.2%

Podemos perceber que, quanto maior o tamanho do arquivo, maior a sua taxa de compressão. Podemos associar isso ao fato de que, quanto maior a repetição de termos no texto, maior será a compressão possível.