Relatório TP2
Fernando Vilela Brandão 2021421907

1-Implementação

Foi implementada uma Árvore de prefixos Trie em Python. Para isso foram utilizadas duas classes TrieNode e Trie:

TrieNode:

Letra que representa a letra do nó em questão

idx que representa um numero identificador para o nó

filhos uma lista que armazena os nós filhos do nó em questão

Trie:

root que armazea um TrieNode vazio

next_idx que armeza o index do próximo nó a ser a dicionário

dict uma lista que funciona como um dicionário

max_int o maior valor armazenado

search(word): tem como parametro uma palavra que será buscada na árvore e retornando o nó do maior prefixo na árvore

insert(word): insere os caracteres na árvore da seguinte forma vai procurando prefixos ao analisar cada caracter se não encontrar adiciona um filho no nó current caso contrário ao processar mais uma letra e encontre um prefixo current é atualizado para o nó que representa esse prefixo.

Além disso foram criadas funções para facilitar a implementação sendo ela

read_recur(dicionario,idx) Utilizando recursão varre o dicionário e adicionando as letras de cada nó em um string que vai acumulando os prefixos

decompress(dicionario): Pega o dicionário passado com parâmetro e retorna o texto armazenado no dicionário. Esse dicionário armazena tuplas sendo o primeiro elemento um caractere representado pelo nó da árvore e o index do do prefixo que é varrido com o auxilio da função read_recur.

to_bin(dicionário,out_file): para cada tupla armazenada no dicionário é feito o encode do caracter no padrão utf-16 (2 bytes) e a conversão do inteiro do nó em um inteiro de 2 bytes e é feita a escrita dos 4 bytes no arquivo out_file

from_bin(in_file): pega os dados codificados e comprimidos decodifica os bytes par a par e gera o dicionário da árvore que ser transformada em texto por meio de decompress(dicionario).

2-Taxa de Compressão

Foram testados em 10 arquivos de teste o algoritmo de compressão

Arquivo	Peso Antes da Compressão	Peso Após Compressão	Taxa de Compressão(%)
1.txt	344,5 kB (344.539 bytes)	257,0 kB (257.040 bytes)	25,4
2.txt	135,8 kB (135.778 bytes)	103,9 kB (103.872 bytes)	23,5
3.txt	159,9 kB (159.948 bytes)	119,5 kB (119.548 bytes)	25,2
4.txt	179,5 kB (179.453 bytes)	129,4 kB (129.384 bytes)	27,9
5.txt	177,0 kB (177.048 bytes)	131,8 kB (131.764 bytes)	25,6
6.txt	194,6 kB (194.610 bytes)	142,4 kB (142.368 bytes)	26,9
7.txt	46,6 kB (46.570 bytes)	39,5 kB (39.464 bytes)	15,3
8.txt	79,8 kB (79.788 bytes)	65,0 kB (64.956 bytes)	18,5
9.txt	352,3 kB (352.296 bytes)	235,1 kB (235.104 bytes)	33,2
10.txt	235,8 kB (235.817 bytes)	156,3 kB (156.256 bytes)	33,7