Algoritmos II - TP 1 - Documentação

Iliana Noronha

2023-1 UFMG

1 Resumo do algoritmo

A ideia por trás do método implementado LZ78 é de comprimir o arquivo passado por meio de chaves. Tanto para a compressão quanto para a descompressão, leremos um arquivo de entrada e escreveremos em um de saída. Sendo assim, para comprimir um arquivo, iniciamos com uma chave 0, que é vazia. Lemos o texto passado e verificamos se a substring que está sendo lida já ocorreu anteriormente no texto. Caso isso seja verdadeiro, passamos ao próximo caracter. Caso contrário, será necessário criar uma nova chave, que será exatamente a chave que se repete acrescida do novo caracter (a chave que se repete pode ser a própria chave vazia). Repetiremos o processo até que o arquivo de entrada chegue ao fim. Para a descompressão, basta criar um vetor tal que suas posições representem os indexes. Ele armazenará as strings correspondentes e, ao ler o arquivo de entrada, irá escrever no arquivo de saída as strings relacionadas, descomprimindo o arquivo.

Em outras palavras, para comprimir, seguimos o seguinte passo a passo:

a. Começa-se com um dicionário que só contém a string vazia

b. Procura-se o maior casamento entre as palavras já no dicionário e o início da string ainda não codificada

c. Escreve-se o index do casamento até o ”erro” concatenado com o próximo caracter

d. Guarda-se esse novo símbolo no dicionário (caso ele não seja o fim do arquivo)

e. Repete-se o item b.

2 Implementação

2.1 Função de compressão

A função de compressão é baseada em uma árvore trie tradicional. Cada nó da árvore possui 3 informações: o seu index no dicionário, seu conteúdo e um vetor de ponteiros, que aponta para cada um de seus filhos. Inicialmente, é declarada a raiz que tem seu index igual à 0, e seu conteúdo é vazio.
Para cada caracter c lido, verificamos se já existe esse padrão no texto, ou seja, se o nó que estamos analisando possui c como filho. Caso encontrarmos, apenas atualizamos o nó atual como sendo esse filho e caso não exista, devemos criar um novo nó como sendo esse caracter e defini-lo como um novo filho do nó atual. Nesse caso, após essa inserção, o nó atual volta a ser a raiz, para reiniciarmos a busca. Ao terminarmos o arquivo, imprime-se o index do nó atual e um sinalizador de fim de arquivo.
Para otimizar mais a compressão e a descompressão, foi criada uma variável que armazena quantos bytes deve-se usar para representar os indexes. Ao serem impressos no arquivo de saída, eles são representados

como char. Logo, esse número se inicia com 1, mas caso sejam criados 2^8 nós, torna-se necessário usar 2 bytes para representar os indexes e assim por diante (caso tenha-se criado 2^{8n} nós (com n o número de bytes usados até o momento), aumenta-se n em uma unidade). Foi preciso fazer essa transformação de char para int para reduzir o espaço gasto, dado que um caracter ocupa o espaço de 1 bit somente.

2.2 Função de descompressão

Para a descompressão, foi criado um vetor de string para representar o dicionário. Ao lermos o arquivo de entrada, armazenamos o index passado e o char correspondente. Buscamos no dicionário aquele índice e concatenamos com o caracter lido. Com isso, formamos uma substring que escreveremos no arquivo de saída e adicionaremos em nosso dicionário. Repetiremos esses passos até que todo o arquivo seja lido.
Note que nessa função, a variável que armazena os bytes usados é importante para que possamos (re)traduzir corretamente o index de um número armazenado como char para um int novamente.

2.3 Main

A main é bem simples: inicialmente ela faz uma análise de quantos argumentos estão sendo passados. Caso sejam 3, não é necessário alterar o nome do arquivo de entrada, apenas a extensão. Sendo assim, o nome do arquivo de saída será o nome do arquivo de entrada e caso queiramos comprimir, substituimos o ".txt" por ".z78" e chamamos a função de compressão e, caso queiramos descomprimir, o ".z78" torna-se ".txt" e chamamos a função de descomprimir. Alternativamente, caso o número de argumentos seja 5, o arquivo de saída será o quinto argumento, e comprimiremos ou descomprimiremos o arquivo de acordo com o segundo argumento (c ou x).

3 Taxa de compressão

Os casos teste foram retirados do site do Governo Federal e são textos de literatura de domínio público (que podem ser acessados aqui). Apenas o livro os Miseráveis foi retirado do seguinte site: `https://www.agr-tc.pt/bibliotecadigital/aetc/download/658/Os%20Miseraveis%20-%20Victor%20Hugo.pdf`. Esse é um exemplo extra, pois o arquivo possui tamanho maior que 2MB.
A taxa de compressão foi calculada da seguinte forma:

$$\textit{Taxa de compressão (\%)} = 100\left(1 - \frac{\textit{Tamanho do arquivo comprimido}}{\textit{Tamanho do arquivo original}}\right)$$

Em outras palavras, uma taxa de compressão de 0% significa que o arquivo não teve seu tamanho reduzido, isto é, o arquivo original e a compressão possuem mesmo tamanho e a taxa de compressão em 100% é máxima, ou seja, o arquivo comprimido tem tamanho 0 (impossível, por sinal).

Abaixo, as taxas de compressão para os exemplos passados nas instruções do TP assim como os casos teste descritos acima.

a. **A Cartomante**
Tamanho Original: 20 KB
Tamanho Compressão: 15 KB
Taxa de Compressão: 25%

b. **A Cidade e as Serras**
Tamanho Original: 440 KB
Tamanho Compressão: 248 KB
Taxa de Compressão: 44%

c. **A Dança dos Ossos**
Tamanho Original: 42 KB
Tamanho Compressão: 27 KB
Taxa de Compressão: 36%

d. **A Volta ao Mundo em 80 Dias**
Tamanho Original: 437 KB
Tamanho Compressão: 238 KB
Taxa de Compressão: 46%

e. **As Maluquices do Imperador**
Tamanho Original: 325 KB
Tamanho Compressão: 183 KB
Taxa de Compressão: 44%

f. **Carolina**
Tamanho Original: 37 KB
Tamanho Compressão: 24 KB
Taxa de Compressão: 35%

g. **Cervantes, Dom Quixote e outras e-crônicas de nossos tempos**
Tamanho Original: 169 KB
Tamanho Compressão: 102 KB
Taxa de Compressão: 40%

h. **Clara**
Tamanho Original: 866 bytes
Tamanho Compressão: 768 bytes
Taxa de Compressão: 11%

i. **Constituição de 1988**
Tamanho Original: 652 KB
Tamanho Compressão: 290 KB
Taxa de Compressão: 56%

j. **Dom Casmurro**
Tamanho Original: 410 KB
Tamanho Compressão: 231 KB
Taxa de Compressão: 44%

k. **Memórias Póstumas de Brás Cubas**
Tamanho Original: 363 KB
Tamanho Compressão: 201 KB
Taxa de Compressão: 45%

l. **Obras Seletas**
Tamanho Original: 399 KB
Tamanho Compressão: 219 KB
Taxa de Compressão: 45%

m. **O Cortiço**
Tamanho Original: 495 KB
Tamanho Compressão: 276 KB
Taxa de Compressão: 44%

n. **Os Lusíadas**
Tamanho Original: 345 KB
Tamanho Compressão: 195 KB
Taxa de Compressão: 44%

o. **Os Miseráveis - ACIMA DE 2MB**
Tamanho Original: 3,2 MB
Tamanho Compressão: 1,8 MB
Taxa de Compressão: 44%

p. **Quincas Borba**
Tamanho Original: 454 KB
Tamanho Compressão: 258 KB
Taxa de Compressão: 43%

Perceba que a média da taxa de compressão gira em torno dos 40%, sendo menor em casos do arquivo ser muito pequeno (como o caso do poema Clara que possui apenas 866 bytes, o que indica que não podemos reduzir muito o que já temos. Note que, em casos de arquivos que ultrapassam 2MB, a taxa de compressão é parecida com as demais. De qualquer maneira, é possível observar que conseguimos comprimir bem os arquivos.