

Compressão e Expansão de dados com o Algoritmo de Huffman

Leonardo Nunes de Oliveira

¹Ciência da Computação – Centro Universitário IESB
Asa Sul – Brasília/DF – Brasil

ti.userleo@hotmail.com

Abstract. *This document presents a scope of a project for academic purposes of the course Integrator Project I of the Computer Science course that will be done in the Java programming language, using the Huffman Algorithm data structure for data compression (compression) and data expansion (decompression) , List, Binary Tree and Map.*

Resumo. *Este documento apresenta um escopo de um projeto para fins acadêmicos da disciplina Projeto Integrador I do curso Ciências da Computação que será feito na linguagem de programação Java, utilizando da estrutura de dados Algoritmo de Huffman para compressão(compactação) e expansão(descompactação) de dados, Lista, Árvore binária e Map.*

1. Introdução

Em 1950, David A. Huffman como um dos grandes pioneiros da Ciência da Computação, enquanto estudante de doutorado desenvolveu um método de compressão/expansão de dados. Compressão e expansão de dados é colocar dados contendo informações em um espaço menor de armazenamento e de mais rápido acesso. Por exemplo, se esse método for utilizado em cima de um texto "ana adora banana", o compressor pode utilizar da repetição com mais frequência de palavras como o "a" e o "n" convertendo para bits que ocupariam menos espaço se comparado ao tamanho do mesmo caractere na tabela ASCII, que mostra o tamanho em bits de cada caractere em seu estado original. Com o passar dos anos, a quantidade de informações no mundo vêm crescendo muito e com isso necessita-se de mais espaço de armazenamento e tempo para ler essas informações. O método de compressão/expansão de dados desenvolvido por David A. Huffman foi denominado de Algoritmo de Huffman. O algoritmo foi originado especificamente para operar em cima de dados em forma de texto, mas com o passar dos anos foi utilizado também em cima de outros formatos de dados, como imagens e áudio. Do Algoritmo de Huffman foram originados vários programas que operam utilizando compressão e descompressão de dados, como por exemplo alguns dos mais famosos hoje WinRAR e WinZIP, fazendo com que o usuário possa ter um menor espaço de armazenamento e uma leitura mais rápida de seus dados.

2. Objetivo geral

Desenvolver um programa que realize a compressão/descompressão de dados com maior velocidade e menor espaço de armazenamento possível utilizando da estrutura de dados Algoritmo de Huffman, Lista, Árvore binária e Map no ambiente de programação Java.

3. Objetivo específico

Para desenvolver o programa proposto, as estruturas serão feitas manualmente sem a importação de bibliotecas. A seguir os procedimentos da execução do programa:

3.1. Frequência

Inicialmente, o programa receberá como entrada um arquivo de texto contendo o texto a ser compactado, e logo em seguida irá percorrer o texto contando as ocorrências de cada caractere. A frequência de cada caractere será armazenada em uma Lista.

3.2. Árvore binária

Em cima das frequências de cada caractere armazenadas na Lista, o programa vai montar uma Árvore binária da seguinte forma: no início o algoritmo vai pegar sempre as duas primeiras ocorrências da lista, que serão os caracteres com menor frequência no texto (na Lista estarão ordenadas das frequências menores para as maiores), e com elas vai fazer um galho da árvore colocando na folha da esquerda o bit "0" para o caractere com menor frequência e na outra o bit "1" para o com maior frequência, e no nó do galho feito será armazenado a soma das duas frequências dos caracteres das duas folhas. Em seguida, o nó gerado voltará para a lista seguindo a ordem do menor para o maior, tendo seu valor como sua frequência. O procedimento se repetirá até não restar nenhum elemento na Lista.

3.3. Compressão

Com a Árvore binária feita, o algoritmo vai percorre-la fazendo a compressão dos dados de suas folhas, quanto mais repetido for o caractere no texto original, menos o programa terá que percorrer a árvore para encontra-lo, e com isso menos bits serão usados para seu reconhecimento, e quanto menos repetido for o caractere mais o programa terá que percorrer a árvore, e com isso mais bits serão atribuídos para esse caractere. A árvore será percorrida começando por sua raiz até chegar na folha do caractere, em seguida no caminho que foi percorrido estarão os bits que serão atribuídos para a sua compressão. Depois de codificar um caractere, o programa vai gravar a sequência de bits em um novo arquivo de texto, e a busca recomeçará, o procedimento se repetirá até que o conjunto de caracteres comprimidos seja igual ao conjunto de caracteres do texto original.

3.4. Dicionário de bits

Após a compressão de cada caractere da Árvore binária, sua sequência de bits será armazenada em um dicionário para ser usado posteriormente na descompressão do código, será utilizado da estrutura de dados Map para guardar os bits e seu respectivo caractere.

3.5. Descompressão

Após ter a compressão do texto completa, para realizar a descompressão o programa usará o dicionário de bits para ler a sequência de bits que foi resultante da compressão, atribuindo cada sequência a seu devido caractere, descomprimindo o texto.

4. Referencial teórico

Utilizei dos sites e vídeos na internet a seguir para me auxiliarem na produção deste documento, e também do conteúdo adquirido até o momento da disciplina Estrutura de Dados do curso Ciências da Computação do Centro Universitário de Brasília - IESB.

[Diogo Enes and Alão] [Cipoli] [Godoy] [Abreu]

References

- Abreu, R. Algoritmo de huffman. youtube. 2015. Disponível em <https://www.youtube.com/watch?v=UEqQbF35730t=384s>. Acesso em: 18 Março de 2020.
- Cipoli, P. O que é compressão/compactação de arquivos?. canaltech. Disponível em <https://canaltech.com.br/utilitarios/O-que-e-compressaocompactacao-de-arquivos/>. Acesso em: 12 Março de 2020.
- Diogo Enes, F. D. and Alão, T. M. Algoritmo de huffman. universidade fernando pessoa multimedia. 2020. Disponível em <http://multimedia.ufp.pt/codecs/compressao-sem-perdas/codificacao-estatistica/algoritmo-de-huffman/>. Acesso em: 10 Março de 2020.
- Godoy, V. Codificação de huffman. youtube. 2015. Disponível em <https://www.youtube.com/watch?v=xQQt5myz00o>. Acesso em: 14 Março de 2020.