



COMPRESSÃO E DESCOMPRESSÃO COM O ALGORITMO DE HUFFMAN

Itajaí

2024

RESUMO

O relatório a seguir vai mostrar de que forma funciona o algoritmo de Huffman, sendo ele um dos métodos mais eficientes e amplamente utilizados para a compressão e descompressão de arquivos, imagens, áudio e transmissão de dados, baseado em árvores binárias onde símbolos mais frequentes recebem códigos curtos e símbolos menos frequentes recebem códigos longos, além disso será mostrado como que funciona de forma resumida e demonstrando quais as vantagens e desvantagens do uso dele.

LISTA DE ILUSTRAÇÕES

Figura 1 – Imagem retirada do site BIT IS MITY para usar como referência, acesso: <https://bitismyth.wordpress.com/2016/01/22/codificacao-de-huffman-e-compressao-de-dados/>

Figura 2 – Imagem retirada do site BIT IS MITY para usar como referência, acesso: <https://bitismyth.wordpress.com/2016/01/22/codificacao-de-huffman-e-compressao-de-dados/>

SUMÁRIO

1 INTRODUÇÃO	5
2 COMPRESSÃO DO ALGORITMO DE HUFFMAN.....	6
2.1 FUNDAMENTOS DA COMPRESSÃO.....	6
2.2 CONSTRUÇÃO DA ÁRVORE DE HUFFMAN.....	6
2.3 DESCOMPRESSÃO DO ALGORITMO DE HUFFMAN	9
2.4 VANTAGENS E APLICAÇÕES	10
2.5 DESVANTAGENS DO ALGORITMO DE HUFFMAN.....	11
3 CONCLUSÃO	13
REFERÊNCIAS.....	14

1 INTRODUÇÃO

A compreensão de dados é de grande importância, pois principalmente ela serve para otimização de armazenamentos e transmissões de informações, reduzindo de forma eficiente, quase sem perda significativa de informações. E entre vários algoritmos existentes, o algoritmo de Huffman é um dos melhores por se destacar na sua eficiência e simplicidade de ser feita. Portanto esse relatório irá mostrar o funcionamento de tal algoritmo utilizando como base as pesquisas feitas nos seguintes materiais: "Codificação Huffman" de Paulo Costa (2010), a página da Wikipédia sobre Codificação de Huffman, as aulas de Estruturas de Dados do IME-USP e a página BITISMYTH, Codificação de Huffman e compressão de dados.

2 COMPRESSÃO DO ALGORITMO DE HUFFMAN

2.1 FUNDAMENTOS DA COMPRESSÃO

O algoritmo de Huffman, criado por David A. Huffman em 1952, é um sistema de compressão de dados sem perda significativa que utiliza uma abordagem baseada em árvore para gerar códigos binários de comprimento variável. A ideia em geral é atribuir códigos curtos a símbolos mais frequentes e códigos longos a símbolos menos frequentes, reduzindo assim o tamanho total dos dados codificados.

2.2 CONSTRUÇÃO DA ÁRVORE DE HUFFMAN

1. Contagem de Frequência:

- O primeiro passo é calcular a frequência de ocorrência de cada símbolo no conjunto de dados a ser comprimido. O resultado se resume em uma tabela de frequências. Exemplando como seria mostrada a frequência (retirado do site BIT IN MYTH), a String "EEEEEEEEEE OOOOOO TTTTT MMMM PP CC XX UU SS DD AA L N Q R" seria mostrado dessa forma, ordenada.

char: freq

'L': 1

'N': 1

'Q': 1

'R': 1

'A': 2

'D': 2

'S': 2

'U': 2

'X': 2

'C': 3

'P': 3

'M': 4

'T': 5

'O': 6

': 8

'E': 9

2. Criação dos Nós:

- Cada símbolo e sua frequência são representados como nós em uma fila de prioridade (ou min-heap), onde o nó com menor frequência tem a maior prioridade, tendo um exemplo na Figura 1.

3. Construção da Árvore:

- A árvore de Huffman é construída iterativamente. Dois nós com as menores frequências são removidos da fila, combinados em um nó pai cuja frequência é a soma das frequências dos nós filhos, e este nó pai é reinserido na fila. Esse processo é repetido até que reste apenas um nó na fila, que se torna a raiz da árvore de Huffman, sendo exemplado na Figura 2.

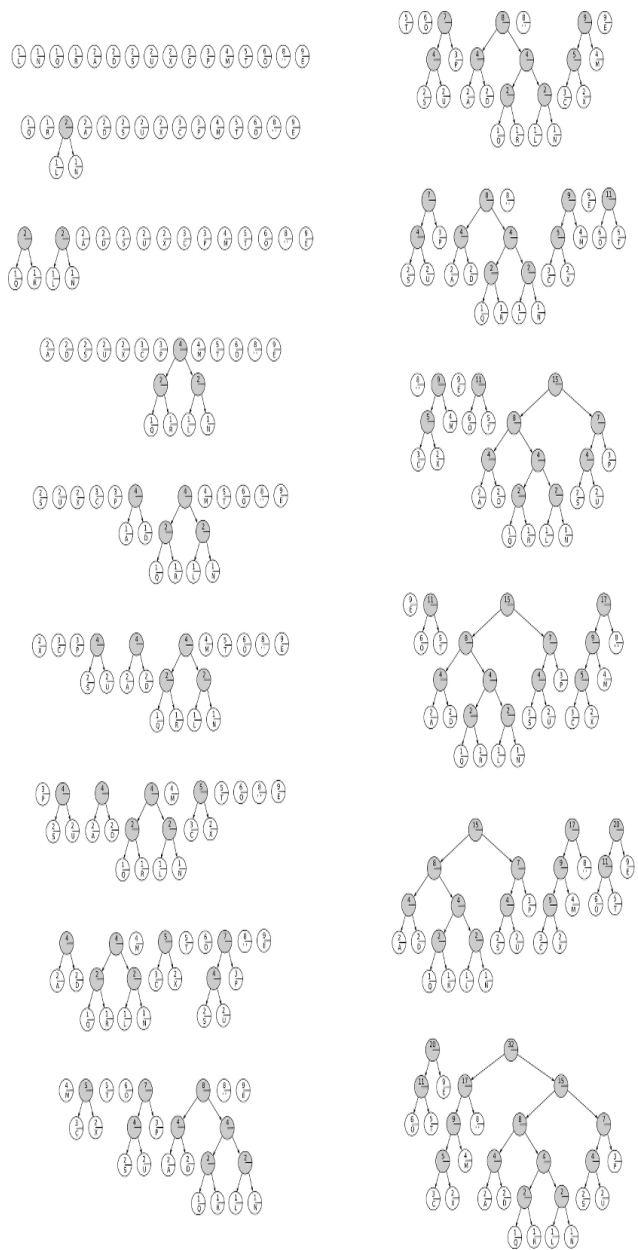


Figura 1 – Imagem retirado como referência, fonte: BIT IS MYTH

Ao ser inserido os caracteres a árvore de Huffman ficara na forma que está a Figura 2.

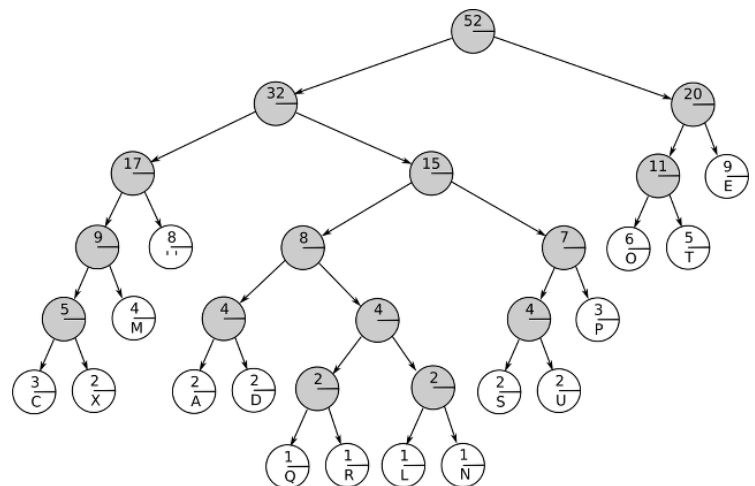


Figura 2 – Imagem retirado como referência, fonte: BIT IS MYTH

4. Geração dos Códigos:

- Os códigos binários para cada símbolo são determinados pela travessia da árvore de Huffman. Sendo redirecionados à esquerda correspondem ao bit '0' e movimentos à direita correspondem ao bit '1'. O código de cada símbolo é formado pelo caminho da raiz até o nó folha correspondente.

2.3 DESCOMPRESSÃO DO ALGORITMO DE HUFFMAN

A descompressão é o processo inverso da compressão. O objetivo é reconstruir os dados originais a partir dos códigos binários gerados pela árvore de Huffman.

1. Reconstrução da Árvore:

- Para descomprimir os dados, a árvore de Huffman usada durante a compressão deve ser conhecida. Frequentemente, a árvore ou as informações necessárias para sua reconstrução são armazenadas junto com os dados comprimidos.

2. Decodificação dos Dados:

- A descompressão começa na raiz da árvore de Huffman. Para cada bit no fluxo de dados comprimido, movemos para a esquerda (bit '0') ou para a direita (bit '1') na árvore. Quando um nó folha é alcançado, o símbolo correspondente é recuperado. Esse processo é repetido até que todos os símbolos comprimidos sejam decodificados.

2.4 VANTAGENS E APLICAÇÕES

Eficiência

O algoritmo de Huffman é altamente eficiente para conjuntos de dados onde certos símbolos ocorrem com maior frequência. A compressão é realizada de maneira a garantir que o tamanho total dos dados comprimidos seja o menor possível, de acordo com a distribuição de frequência dos símbolos.

Aplicações Práticas

O algoritmo de Huffman é amplamente utilizado em diversas áreas, incluindo:

Compressão de Arquivos: Formatos como ZIP e GZIP utilizam variações do algoritmo de Huffman para reduzir o tamanho dos arquivos.

Compressão de Imagens e Áudio: Formatos como JPEG e MP3 utilizam Huffman para codificação de entropia.

Transmissão de Dados: Protocolos de comunicação e armazenamento de dados muitas vezes empregam Huffman para otimizar a eficiência da transmissão.

2.5 DESVANTAGENS DO ALGORITMO DE HUFFMAN

Embora o algoritmo de Huffman seja amplamente utilizado e eficaz para compressão de dados, ele apresenta algumas desvantagens que podem limitar sua aplicabilidade em certos contextos:

1. Necessidade de Conhecimento das Frequências dos Símbolos

Para construir a árvore de Huffman, é necessário conhecer previamente as frequências dos símbolos no conjunto de dados. Isso significa que, para fluxos de dados em tempo real ou para dados que mudam dinamicamente, a eficiência do algoritmo pode ser reduzida, uma vez que é preciso recalcular a árvore de Huffman periodicamente.

2. Ineficiente para Pequenos Conjuntos de Dados

Para conjuntos de dados muito pequenos, a sobrecarga de construir a árvore de Huffman pode ser maior do que os benefícios obtidos pela compressão. Nesses casos, a diferença no tamanho do arquivo comprimido pode ser insignificante ou até mesmo negativa, devido à inclusão das informações da árvore de Huffman.

3. Complexidade de Implementação

Embora o conceito do algoritmo seja simples, a implementação prática pode ser complexa, especialmente em sistemas onde o desempenho é crítico. A construção e a manipulação da árvore de Huffman exigem um entendimento aprofundado das estruturas de dados e dos algoritmos de fila de prioridade.

4. Compressão Estática vs. Compressão Adaptativa

O algoritmo de Huffman tradicional é uma técnica de compressão estática, onde a árvore é construída com base em uma análise inicial dos dados. Em contraste, técnicas de compressão adaptativa, como o Algoritmo de Huffman Adaptativo (AHC), atualizam a árvore dinamicamente conforme novos dados são processados, o que pode ser mais eficiente para fluxos de dados variáveis, mas também mais complexo de implementar.

5. Alternativas Mais Eficientes em Alguns Cenários

Existem algoritmos de compressão mais avançados, como o Lempel-Ziv-Welch (LZW) e o algoritmo de compressão de blocos (BWT), que podem oferecer melhores taxas de compressão em determinados tipos de dados, especialmente quando a redundância não é baseada apenas na frequência de símbolos individuais, mas também em padrões repetitivos e sequências de símbolos.

3 CONCLUSÃO

O algoritmo de Huffman é uma ferramenta poderosa para compressão de dados, proporcionando uma redução eficiente do tamanho dos dados sem perda significativa de informação. Sua aplicação é ampla e essencial em áreas que demandam otimização do uso de recursos de armazenamento e transmissão. A compreensão detalhada do processo de construção e uso da árvore de Huffman é fundamental para implementar e utilizar essa técnica de maneira eficaz.

REFERÊNCIAS

COSTA, P. Codificação Huffman. Universidade Federal do Espírito Santo, 2010. Disponível em: <http://www.inf.ufes.br/~pdcosta/ensino/2010-1-estruturas-de-dados/material/CodificacaoHuffman.pdf>. Acesso em: 26 maio 2024.

Wikipédia. Codificação de Huffman. Disponível em: https://pt.wikipedia.org/wiki/Codificação_de_Huffman. Acesso em: 26 maio 2024.

INSTITUTO DE MATEMÁTICA E ESTATÍSTICA DA USP. Codificação de Huffman. Disponível em: <https://www.ime.usp.br/~pf/estruturas-de-dados/aulas/huffman.html>. Acesso em: 26 maio 2024.

BITISMYTH. Codificação de Huffman e compressão de dados. 2016. Disponível em: <https://bitismyth.wordpress.com/2016/01/22/codificacao-de-huffman-e-compressao-de-dados>. Acesso em: 26 maio 2024.