

# Trabalho 3: algoritmos de compressão

Frederico Hansel dos Santos Gassen

fhgassen@inf.ufsm.br

## 1. Introdução

A codificação de Huffman é um método de compressão criado em 1952 por David A. Huffman comumente utilizado quando não se deseja trabalhar com a perda de dados (lossless data compression). Esse algoritmo baseia-se na frequência ou probabilidade de ocorrência de cada símbolo (ou caractere) em uma entrada onde cada um possui um código, gerando códigos reduzidos para cada um deles. Ela baseia-se no princípio de utilizar um número menor de bits para codificar um dado que aparece mais frequentemente, o que reduzirá o tamanho da saída. Esse princípio é respeitado pelo algoritmo de geração da chamada Árvore de Huffman, cuja organização será explanada posteriormente.

## 2. Algoritmo

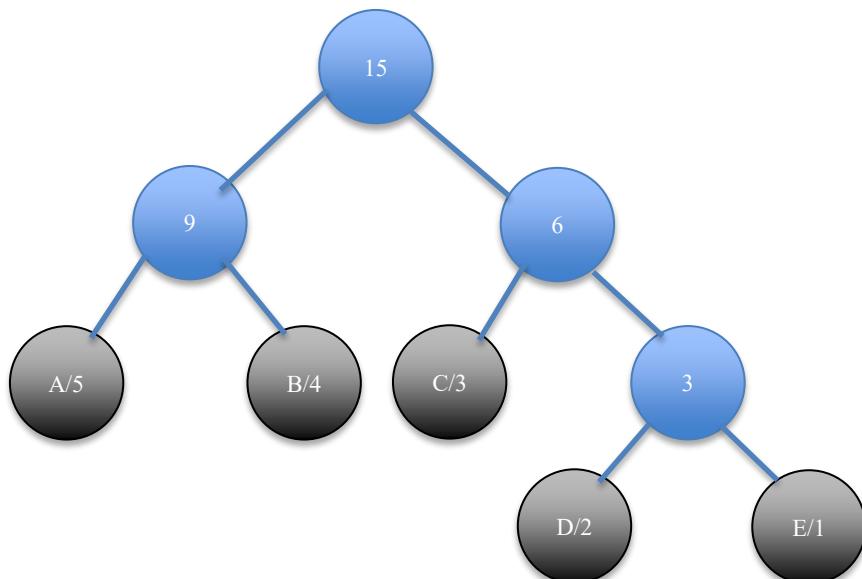
Inicialmente, montamos uma tabela com os caracteres e a frequência de uma entrada. Para melhor visualização, será utilizado como exemplo a entrada AAAAABBBBCCCCDDE:

Caractere	A	B	C	D	E
Código	000	001	010	011	100
Frequência	5	4	3	2	1

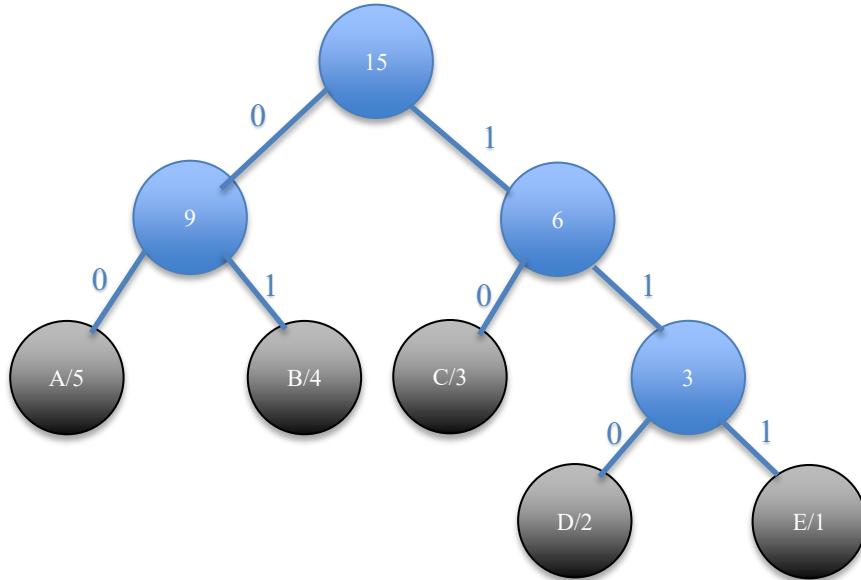
O próximo passo é a montagem da Árvore de Huffman. Para isso, precisamos criar um nó para cada caractere em ordem decrescente da esquerda pra direita:



Em seguida, juntamos os dois nós com menor frequência recursivamente até formarmos uma raiz:



Após, atribui-se 1 às arestas da direita e 0 às arestas da esquerda:



Nesse momento temos nossa Árvore de Huffman concluída e já é possível montar as novas codificações. Para esta montagem, basta apenas seguir os caminhos por arestas até as folhas, acumulando os valores nela armazenados:

Caractere	A	B	C	D	E
Código antigo	000	001	010	011	100
Código novo	00	01	10	110	111

Antes da codificação, a frase original AAAAABBBCCCDDE era codificada como 000000000000000010010010010010010011011100, totalizando 45 bits. Após, codificamos como 00000000000010101011010110110111, totalizando 33 bits, uma redução de 12 bits. Como esse exemplo é pequeno, a redução não foi tão expressiva, porém quando se trabalha com entradas maiores e com frequências enormes, as reduções tornam-se consideráveis.

Para a descompressão da entrada codificada, é essencial que se tenha acesso a Árvore de Huffman, e para isso é necessário que a mesma seja enviada junta para o módulo que deseja decodificar a mensagem. E para decodificá-la, basta começar na raiz da árvore e ir seguindo os bits do começo da codificação. Quando chegar em uma folha, guarda-se o caractere encontrado, volta-se a raiz, e continua-se na decodificação de onde parou.

### 3. Aplicações

A codificação de Huffman é encontrada em várias aplicações de compressão. GZIP, PKZIP, BZIP2 e até JPEG e PNG são exemplos que utilizam o método. Ele também está presente no comando compact, do UNIX. Também há artigos onde mostram a utilização do mesmo em redes neurais, com resultados que indicaram uma redução de 20% a 30% no armazenamento de rede.

## Referências

- MORRIS, John. **Huffman Encoding**. 1998. Disponível em: <<https://www.cs.auckland.ac.nz/software/AlgAnim/huffman.html>>. Acesso em: 16 jul. 2018.
- SOUZA, Jairo Francisco de. **Strings (compressão)**. Disponível em: <[http://www.ufjf.br/jairo\\_souza/files/2012/11/8-Strings-Compress%C3%A3o1.pdf](http://www.ufjf.br/jairo_souza/files/2012/11/8-Strings-Compress%C3%A3o1.pdf)>. Acesso em: 16 jul. 2018.
- HAN, Song; MAO, Huizi; DALLY, William J.. **DEEP COMPRESSION: COMPRESSING DEEP NEURAL NETWORKS WITH PRUNING, TRAINED QUANTIZATION AND HUFFMAN CODING**. 2016. Disponível em: <<https://arxiv.org/pdf/1510.00149.pdf>>. Acesso em: 16 jul. 2018.
- SHARMA, Mamta. **Compression Using Huffman Coding**. 2010. Disponível em: <<https://pdfs.semanticscholar.org/89c2/03919ce446762d9b768ac0974f54021f338e.pdf>>. Acesso em: 16 jul. 2018.
- CODIFICAÇÃO de Huffman. Disponível em: <[https://pt.wikipedia.org/wiki/Codificação\\_de\\_Huffman](https://pt.wikipedia.org/wiki/Codificação_de_Huffman)>. Acesso em: 16 jul. 2018.