**Algoritmo de Huffman**

**Juliano Leonardo Soares**

jlsoares@inf.ufsm.br

**⦁ Introdução**

A codificação de Huffman, é um algoritmo para a compressão de arquivos, principalmente arquivos texto, neste tipo de codificação o algoritmo tem uma técnica para compressão de dados que envolve encontrar uma codificação para o alfabeto atribuindo um código de bits de comprimento variável a cada letra ou símbolo, onde os símbolos com menos probabilidade tem um código de bits de comprimento maior e os símbolos que têm maior probabilidade com menor comprimento, tornando assim a codificação de Huffman umas das codificações mais eficientes.

O algoritmo foi desenvolvido independentemente por Faller (1973) e Gallager (1978), sendo melhorado por Knuth (1985), ficando assim conhecido como algoritmo FGK. Existe ainda outro melhoramento concebidos por Vitter (1987), ficando conhecido por algoritmo

**⦁ Algoritmo**

O Algoritmo Huffman cria uma árvore ordenada com todos os símbolos e a frequência com que aparecem. As pontas da árvore, ou seja, Os galhos são construídos começando com os símbolos que menos aparecem, de menor frequencia. Sucessivamente, os dois símbolos de mais baixa frequência de aparecimento são retirados da lista e unidos a um núcleo cujo peso vale a soma das frequências dos dois símbolos. O símbolo menor ou como no meu exemplo que são caracteres do alfabeto o mais próximo de A é atribuído à ramificação 1, o outro à ramificação 0 e assim por diante, considerando cada núcleo formado como um novo símbolo, até obter um só núcleo parente, chamado raiz.

É extremamente simples implementa-lo, são poucos passos para a construção da árvore binária de símbolos, em que são aplicadas tabelas auxiliares.

1° Ordenação dos símbolos por ordem de frequência: Vejamos a seguinte palavra “BANANA”. Veja as frequências de aparecimentos das letras: B = 1, A = 3, N = 2.

2° pegamos os dois caracteres ou símbolos com a menor frequência, no meu exemplo é o B e o N cada um com suas respectivas frequências 1 e 2. eles serviram de base ou galho de minha árvore com as seguintes posições B antes do N, ou seja, mais à esquerda na arvore por ser um símbolo menor que N e após cria-se um nó (núcleo) acima deles na arvore com o peso da soma das frequências dos dois símbolos.

3° acrescenta-se o próximo caractere ou símbolo de maior frequência no meu exemplo é o A e cria um novo nó que aponta para o símbolo recém adicionado e ao antigo nó com referência ao peso da soma das frequências dos dois símbolos anteriormente adicionados. Resultando na seguinte árvore:

b 1 00 n 2 01 a 3 1



E por fim gerando o código binário pala a palavra banana 001011011 que deve ser decodificado por um decodificador huffman para poder obter o dado novamente.

**⦁ Aplicações**

As aplicações do algoritmo são inúmeras, pois ele sendo um algoritmo que permite otimizar a informação sem qualquer tipo de perdas, ele é amplamente utilizado em aplicações de compressão, que vão de GZIP, PKZIP, BZIP2 a formatos de imagens como JPEG e PNG, MPEG (vídeo), em codificações de áudio, como CD ou parte da recente norma MP3, utilização do codec de Huffman nas normas de fax.

**Referências**

<http://multimedia.ufp.pt/codecs/compressao-sem-perdas/codificacao-estatistica/algoritmo-de-huffman-adaptativo/>

<https://www.ime.usp.br/~pf/analise_de_algoritmos/aulas/huffman.html>

<http://professor.ufabc.edu.br/~francisco.massetto/ni/Aula6.pdf>

<https://pt.wikipedia.org/wiki/Codifica%C3%A7%C3%A3o_de_Huffman>

<https://br.ccm.net/contents/729-codificacao-de-huffman>