

Lista 03 - Algoritmo de Huffman

Professor: Fábio Engel de Camargo
Disciplina: Sistemas Multimídia e Hiperemídia
Meio para entrega: Moodle

João Marcos 2264501

- Qual a diferença entre código de comprimento fixo e código de comprimento variável? Por que código de comprimento variável precisa ser livre de prefixos? Código de comprimento fixo precisa ser livre de prefixos?

R: Código de comprimento fixo e código de comprimento variável são duas formas de representação de dados binários em que cada símbolo é codificado por uma sequência de bits. No código de comprimento fixo, cada símbolo é representado por um número fixo de bits, enquanto no código de comprimento variável, os símbolos mais frequentes são representados por códigos mais curtos e os menos frequentes por códigos mais longos. Um código de comprimento variável precisa ser livre de prefixos para evitar ambiguidades na decodificação, enquanto um código de comprimento fixo não precisa ser livre de prefixos, embora em algumas situações seja benéfico.

- Quais das tabelas de códigos abaixo são livres de prefixos?

A	0	0	1	1
B	100	1	01	01
C	10	00	001	001
D	11	11	0001	000

- Uma palavra foi codificada usando o código de Huffman, tendo-se obtido a sequência binária: 1 0 1 1 1 0 1 1 0 1 0 1 1 1 0 0 1 1 1 0 1 0 0. O alfabeto original era constituído pelas letras A, B, C, D, E, I, L, R e T e a letra I foi codificada como “00”. Supondo que estas letras ocorriam com as probabilidades:

P(A) = 0.26 P(D) = 0.01 P(L) = 0.01
P(B) = 0.09 P(E) = 0.07 P(R) = 0.23
P(C) = 0.08 P(I) = 0.22 P(T) = 0.03

- Qual terá sido a palavra codificada?

R: ACERTEI

- Calcule o número médio de bits por carácter obtido pelo uso da codificação de Huffman e compare com a utilização de um código binário de tamanho fixo (8 bits por caractere) para representação do mesmo alfabeto.

R: 9 caracteres, 4 bits por carácter ($2^4 = 16$), Após Huffman houve uma redução de 33,75%.

- Qual a representação (em binário) da árvore binária deste exercício?

R: 1 0 1 1 1 0 1 1 0 1 0 1 1 1 0 0 1 1 1 0 1 0 0

4. Um alfabeto com 256 caracteres podem ser codificados usando o algoritmo de Huffman utilizando quantos bits? Dica: no moodle está uma implementação do algoritmo de Huffman em Python.

R: No máximo 8 bits por carácter