

Lista 03 - Algoritmo de Huffman

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

André Luis Quiosi RA: 2369958

1. 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: os códigos de comprimento fixo definem uma quantidade de bits para representar cada símbolo e os códigos variável definem a quantidade de bits dependendo da frequência do símbolo e não pode ter prefixos por poder causar erros e de-codificar símbolos errados

2. 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

R: nenhuma das tabelas a seguir é livre de prefixos.

3. 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 caracter 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: 3,88 bits

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

001 I 1 R 01 A 00001 D 1 L 1 T 1 E 01 C 1 B

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: de acordo com o código em python, podem ser codificados para 2048bits