PROBLEMA PROPOSTO PARA QUALIFICAÇÃO À VAGA PARA PROGRAMADOR C NA INFORMATEC

INSTRUÇÕES

Caso tenha alguma dúvida entre em contato via e-mail usando o destinatário e assunto mostrados abaixo:

- `Destinatário: vagas@informatec.com.br`
- `Assunto : [Dúvida]Codificação Huffman`

É importante que o assunto acima não seja alterado, pois é usado por um sistema automático de triagem na Informatec (recomendamos copiá-lo e colá-lo no e-mail).

Algumas explicações neste documento encontram-se propositalmente em inglês, haja vista ser o domínio deste idioma condição necessária para qualificação para a vaga de programador na Informatec.

Recomendamos ler este documento até o final.

OBJETIVOS DESTE TESTE

Serão verificados neste teste:

- capacidade de solução de um problema simples, mas real.
- capacidade de implementação de uma solução para o problema (não necessariamente a MELHOR solução, mas sim UMA solução).
- qualidade da documentação da solução. Por exemplo, inteligibilidade do código-fonte, padronização na nomenclatura das variáveis.
- capacidade de entendimento de um texto simples em inglês.

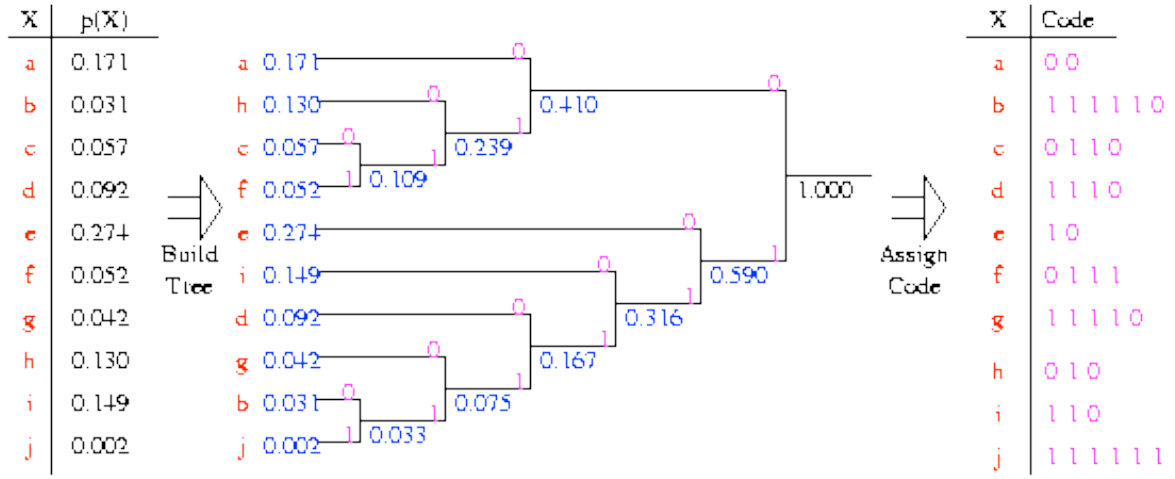
A CODIFICAÇÃO HUFFMAN

The basic idea in Huffman coding is to assign short codewords to those input blocks with high probabilities and long codewords to those with low probabilities.

A Huffman code is designed by merging together the two least probable characters, and repeating this process until there is only one character remaining. A code tree is thus generated and the Huffman code is obtained from the labeling of the code tree. An example of how this is done is shown below.

X	p(X)
a	0.171
b	0.031
c	0.057
d	0.092
e	0.274
f	0.052
g	0.042
h	0.130
i	0.149
j	0.002

The final static code tree is given below:



- *It does not matter how the characters are arranged. I have arranged it above so that the final code tree looks nice and neat.*
- *It does not matter how the final code tree are labeled (with 0s and 1s). I chose to label the upper branches with 0s and the lower branches with 1s.*
- *There may be cases where there is a tie for the two least probable characters. In such cases, any tie-breaking procedure is acceptable.*
- *Huffman codes are not unique.*

PROBLEMA A SER RESOLVIDO

Você deve escrever um programa em C que resolva o problema mostrado abaixo.

Dado que existe um vetor `ENTRADA` de 256 elementos (0 a 255) tal que cada elemento contém um valor (0<=valor<=0xFFFF), crie um segundo vetor `SAIDA`:

```
struct tCodigo {
     unsigned int uiCode;
     int          iBitsUsadosDaEsquerdaParaDireita;
};
unsigned int ENTRADA[256];
tCodigo      SAIDA[256];
```

Como exemplo, usando a tabela mostrada acima, seu programa deveria calcular os seguintes resultados para 3 dos elementos do vetor `SAIDA`:

```
SAIDA[97].uiCode = 0x0000;
SAIDA[97].iBitsUsadosDaEsquerdaParaDireita = 2;

SAIDA[98].uiCode = 0xF800;
SAIDA[98].iBitsUsadosDaEsquerdaParaDireita = 6;

SAIDA[99].uiCode = 0xA000;
SAIDA[99].iBitsUsadosDaEsquerdaParaDireita = 4;
```

Não pode ser usada nenhum biblioteca de terceiros para solução deste problema, e somente podem ser incluídos:

```
#include <ctype.h>
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
```

PRAZO PARA RESOLUÇÃO DO PROBLEMA

Você tem 15 dias úteis a partir da data em que recebeu o e-mail contendo ESTE documento para enviar e-mail para o endereço mostrado acima indicando ter terminado o programa.

IMPORTANTE: não envie seu código-fonte no e-mail. Basta apenas nos avisar que já o terminou. Acreditamos em você!

Então, entraremos em contato via e-mail para agendar uma entrevista na Informatec. Nesta ocasião você nos apresentará seu código-fonte e nós o testaremos usando Borland C++ Builder 6.

ALGUNS AUXÍLIOS

- Bibliografia recomendada:
  The Data Compression Book
  by Mark Nelson and Jean-loup Gailly , M&T Books, New York, NY 1995
  ISBN 1-55851-434-1
  2nd edition

- Links:
  http://www.data-compression.com/lossless.shtml
  http://www.fadden.com/techmisc/hdc

- Tabela ASCII