Compressão de Huffman

Resultado do trabalho de doutorado de David Huffman, este método permite construir uma árvore de codificação que ao substituir caracteres do arquivo original, faz com que ele seja comprimido. O truque empregado é reservar configurações binárias menores em comprimento para os caracteres que são mais frequentes.

No exercício de hoje, o objeto da compressão vai ser uma imagem BMP mapeada. O objetivo deste exercício é mostrar como construir uma árvore de Huffman de grande porte (já que deverá haver cerca de 250 níveis distintos de cor na imagem) e também mostrar como se obtém grandes níveis de compressão ao fazê-la sobre imagens digitais.

Tal redução de tamanho deve-se ao fenômeno denominado "localidade de referência espacial"que sugere que a probabilidade de pixels vizinhos terem o mesmo nível de cor é bastante alta. Por exemplo, numa paisagem, o céu tem grande probabilidade de compartilhar niveis de cor bastante próximos (ou até iguais). Numa foto de pessoa em close-up, a cor de sua pele tende a se concentrar em poucos níveis de cor.

As etapas a desenvolver neste exercício, são as seguintes

ler a imagem Deve-se criar um programa capaz de ler a imagem, importando a matriz de pixels para dentro da memória. Lembre-se que o arquivo a ler seguirá o padrão BMP de imagens mapeadas. Neste tipo de imagem, após os 54 bytes iniciais (que caracterizam o bloco de controle do arquivo), seguem-se 1024 bytes que são a tabela de cores. (embora fossem necessários apenas $256 \times 3 = 768$ bytes, o padrão reserva 1024 por questões de eficiência). Finalmente, depois deste ponto começam os pontos da imagem, à razão de 1 byte por pixel.

A imagem lida sempre terá 512 x 512 pixels. O número de colunas em particular é múltiplo de 4, o que determina que não haverá *stuffing columns*, e simplificará o processamento.

A compressão vai se dar apenas sobre os dados da imagem. Assim o bloco de controle (54 bytes) e a tabela de cores (1024 bytes) não serão manuseados nesta folha.

O resultado final deste procedimento é uma matriz numérica de 512 linhas por 512 colunas. Cada item da matriz é um único caracter, mas aqui tratado como um número inteiro entre 0 e 255.

montar o histograma O próximo passo é contar quantas ocorrências existem de uma determinada cor.

desconsiderar zeros Devem ser ignorados eventuais níveis de cor que tenham 0 ocorrências. Esta etapa não é obrigatória, mas simplifica os cálculos a seguir.

construir a árvore Antes de estudar o algoritmo, vai-se estudar o significado das principais variáveis envolvidos.

A variável *mt* é uma matriz de *n* linhas (depende da variedade de cores da imagem) por 2 colunas. Na primeira coluna está a cor (um número entre 0 e 255) e na segunda coluna está a quantidade de pixels que têm esta cor. A variável *mt* tem limites verticais fluídos (começando em ini e terminando em fim).

A variável arvore é a própria. Contém 3 colunas, sendo a primeira o endereço do filho esquerdo, a segunda o endereço do filho direito e a terceira o conteúdo do nodo. Neste último, valores negativos correspondem a um nível de cor (em valor absoluto) e valores positivos correspondem a nodos intermediários (pelo seu endereço).

```
 1: mt[;1] ← -mt[;1] // negativa coluna 1
 2: iar ← 1
 3: i ← 1
 4: ini ← 1
 5: fim ← número de linhas de mt
 6: enquanto fim-ini > 2
 7:     j ← ini
 8:     men ← +99999
 9:     enquanto j ≤ fim
10:         se mt[j;2] < men
11:             men ← mt[j;2]
12:             imen ← j
13:         fim{se}
14:         j ← j + 1
15:     fim{enquanto}
16:     swap(mt[imen;],mt[ini;])
17:     ini ← ini + 1
18:     j ← ini
19:     men ← +99999
20:     enquanto j ≤ fim
21:         se mt[j;2] < men
22:             men ← mt[j;2]
23:             imen ← j
24:         fim{se}
25:         j ← j + 1
26:     fim{enquanto}
27:     swap(mt[imen;],mt[ini;])
28:     ini ← ini + 1 {ini-2 é o menor, ini-1 é o
        2.menor}
29:     arvore[iar;1] ← i
30:     arvore[iar;2] ← mt[ini-2;1]
31:     arvore[iar;3] ← mt[ini-1;1]
32:     iar ← iar + 1
33:     fim ← fim + 1
34:     mt[fim;1] ← i
35:     mt[fim;2] ← mt[ini-2;2]+mt[ini-1;2]
36:     i ← i + 1
37: fim{enquanto}
```

☞ Para você fazer

Leia o arquivo de nome `CUR 19 .BMP`

Neste arquivo, monte a árvore de Huffman, conforme descrito nesta folha.

Montada a árvore, imprima-a e responda as 4 perguntas a seguir:

1. Quantos pixels tem a cor mais frequente na imagem ? _____

2. Quais os 3 nodos na árvore se esta for for considerada uma matriz e tiver suas linhas numeradas (lembre-se que a raiz é o último nodo desta matriz). Considere as linhas de números. O valor inicial é 1.

79		
116		
167