## Universidade Tecnológica Federal do Paraná

Código de Huffman

Eduardo Vanderlei dos Santos Junior

Juliana Sandis Cavalcante

junho

2017

1. Introdução

O código que foi estudado neste trabalho foi o código de Huffman. O algoritmo foi desenvolvido em 1952 por David A. Huffman que era estudante de doutorado no MIT.

O resultado do algoritmo de Huffman pode ser visto como uma tabela de códigos de tamanho variável para codificar um símbolo da fonte. Assim como em outros métodos de codificação, os símbolos mais comuns são geralmente representados usando-se menos dígitos que os símbolos que aparecem com menos frequência.

A codificação de Huffman é amplamente utilizada em aplicações de compressão, que vão de GZIP, PKZIP, BZIP2 a formatos de imagens como JPEG e PNG. Isso ocorre pela codificação ser simples, rápida e, principalmente, não ter patente. Consequentemente, não se pagam royalties para utilização comercial de aplicações que empregam este método.

1. Codificação de Huffman

A codificação de Huffman é um método de compressão que usa as probabilidades de ocorrência dos símbolos no conjunto de dados a ser comprimido para determinar códigos de tamanho variável para cada símbolo.

Essa atribuição é realizada através de uma árvore binária onde as folhas representam os símbolos presentes nos dados, associados com suas respectivas probabilidades de ocorrência. Os nós intermediários representam a soma das probabilidades de ocorrência de todos os símbolos presentes em suas ramificações e a raiz representa a soma da probabilidade de todos os símbolos no conjunto de dados. O processo se inicia pela junção dos dois símbolos de menor probabilidade, que são então unidos em um nó ao qual é atribuída a soma de suas probabilidades.

A cada aresta da árvore é associado um dos dígitos binários (*0* ou *1*). O código correspondente a cada símbolo é então determinado percorrendo-se a árvore e anotando-se os dígitos das arestas percorridas desde a raiz até a folha que corresponde ao símbolo desejado.

1. Heap

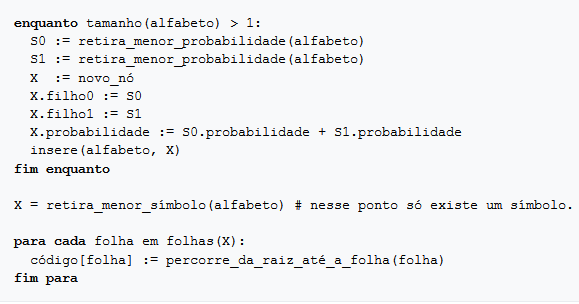
Um heap (binário) é uma estrutura de dados que pode ser vista como uma árvore binária completa. Pode funcionar como uma fila de prioridades.

Cada nó da árvore corresponde a um elemento do vetor que armazena o valor do nó. A árvore é completa em todos os níveis com exceção do nível mais baixo, o qual é completado da esquerda para a direita.

Para que um vetor A seja heap, cada nó i, com exceção da raiz, deve satisfazer a seguinte propriedade: o valor de cada nó é no máximo o valor do seu pai. Essa regra é muito importante na codificação de Huffman , já que nós com uma frequência maior deve ser pai de nós com frequência menores.

1. Algoritmo

Abaixo temos um pseudo-código das etapas do código de Huffman.



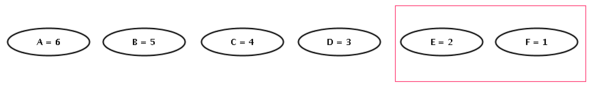
1. Exemplo

Para demonstrar o funcionamento do algoritmo usaremos a sequência de caracteres “AAAAAABBBBBCCCCDDDEEF “ . Na formatação normal essa sequência ocuparia 63 bits.

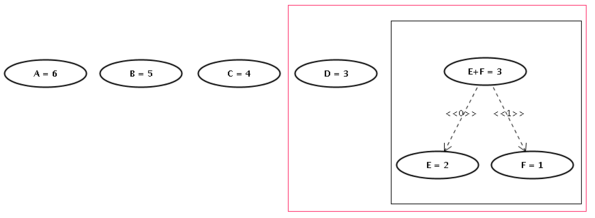
Usando a codificação de Huffman precisamos separar as frequências das letras do conjunto.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Caracteres | A | B | C | D | E | F |
| Contagem | 6 | 5 | 4 | 3 | 2 | 1 |

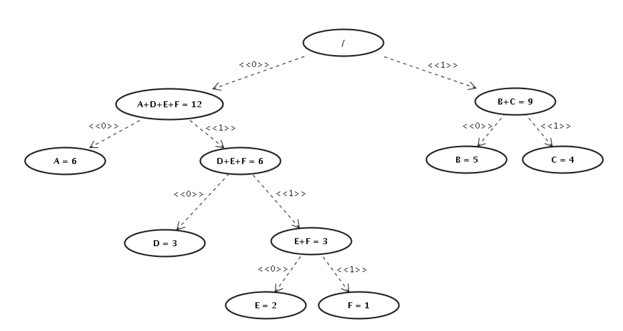
Agora vamos montar a árvore unindo as letras com menores frequências, no caso E e F.



O novo nó criado tem a união das frequências das duas letras unidas e seus filhos são as próprias letras, Também é atribuído para as arestas o valor binário (0 e 1).



O processo vai se repetindo até todas as raízes se unirem, assim teremos uma árvore que definirá os novos códigos para as letras.



Os novos códigos serão definidos pelo caminho realizado para chegar da raiz até as folhas, com os valores binários atribuídos às arestas. Assim temos um tabelas de códigos.

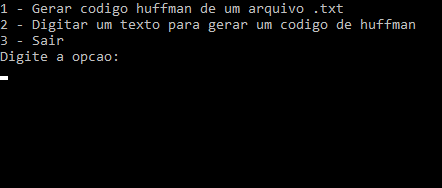
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Caracteres | A | B | C | D | E | F |
| Código | 00 | 10 | 11 | 010 | 0110 | 0111 |

Assim o conjunto que inicialmente tinha uma quantidade de 63 bits foi reduzido para 51 bits a compreensão foi de 20%.

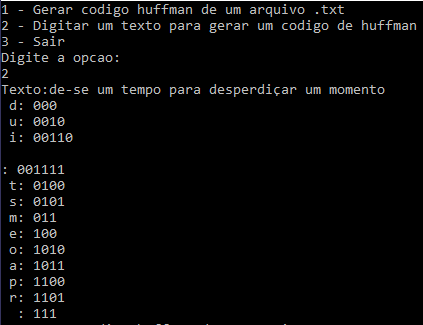
O processo de descompressão é simplesmente uma questão de se traduzir a palavra-código de volta para os caracteres. Isso é geralmente realizado percorrendo a árvore do código de nó em nó, conforme a orientação dada por cada bit lido da palavra-código, até encontrar-se uma folha. Ou seja, o processo resume-se a voltar, por meio do caminho na árvore, para a folha associada ao símbolo original.

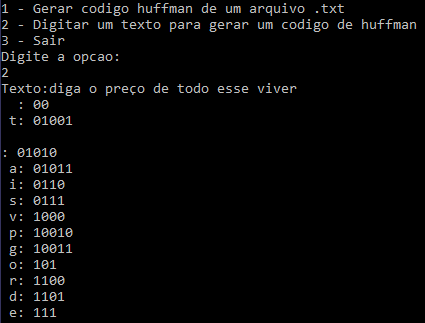
1. Testes

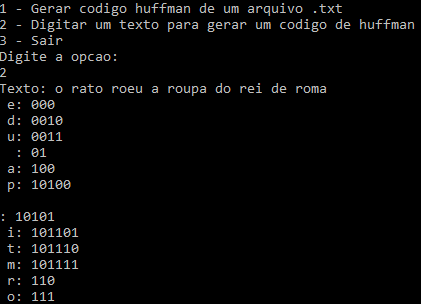
tela inicial do programa:



gerando um código Huffman para um texto:







1. Conclusão

Em casos reais a compressão obtida pode ser bem maior pois a frequência de alguns símbolos é bastante grande enquanto a de outros é quase nula (as vogais "a", "e", "i", "o" e "u" ocorrem com bastante frequência, enquanto algumas letras como "y", "k" ou "z" aparecem bem menos). Essa diferença grande nas frequências faz com que os caracteres mais frequentes, e consequentemente com símbolos mais curtos, sejam melhor representados aumentando bastante a taxa de compressão.