

UNIVERSIDADE FEDERAL DE OURO PRETO
Faculdade de Ciência da Computação

Trabalho Prático 3

Marcus Vinícius Souza Fernandes 19.1.4046
Gabriel Mace dos Santos Ferreira 19.14013

Ouro Preto - MG
Dezembro de 2019

Trabalho Prático 3

Marcus Vinícius Souza Fernandes
Gabriel Mace dos Santos Ferreira

Trabalho Prático do Curso de Ciência da
Computação da Universidade Federal de Ouro
Preto.

INTRODUÇÃO

Inicialmente foi proposta a implementação da codificação de Huffman e o uso de uma tabela hash para resolver um problema concebido pelos próprios alunos. A elaboração do trabalho se deu por meio da discussão entre os membros do grupo sobre como utilizar as estruturas de dados (árvore e tabela hash) nos problemas apresentados.

Sucintamente, a codificação de Huffman consiste na atribuição de um código binário à símbolos ou caracteres que se deseja comprimir, no entanto os códigos possuem comprimentos distintos sendo que os símbolos mais frequentes possuem códigos menores.

A tabela hash ou tabela de dispersão tem como objetivo facilitar e agilizar a busca em memórias primárias através da associação de chaves à valores, por isso são comumente utilizadas para indexar grandes volumes de informação, por exemplo: vetores associativos, caches e conjuntos.

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. Ele foi desenvolvido em 1952 por David A. Huffman que era, na época, estudante de doutorado no MIT, e foi publicado no artigo "A Method for the Construction of Minimum-Redundancy Codes".

A codificação de Huffman consiste na atribuição de um código binário à símbolos ou caracteres que se deseja comprimir, no entanto os códigos possuem comprimentos distintos sendo que os símbolos mais frequentes possuem códigos menores.

Inicialmente faz-se necessária a contagem do número de repetições de cada símbolo/elemento, a seguir é montada uma árvore binária que agrupa os símbolos de acordo com sua frequência, sucessivamente, os dois símbolos de mais baixa frequência de aparecimento são retirados da lista e unidos a um núcleo cujo peso vale a soma das frequências dos dois símbolos. O símbolo de peso mais leve é atribuído à ramificação 1, o outro à ramificação 0 e assim por diante, considerando cada núcleo formado como um novo símbolo, até obter um só núcleo parente, chamado raiz.

Em sequência, percorre-se a árvore de forma a montar os códigos para cada símbolo, ademais os símbolos com repetições menores por estarem mais próximos das folhas apresentam códigos menores, os dados são então re-codificados usando esse dicionário.

Para ilustrar o funcionamento do método, vamos comprimir a sequência de caracteres `AAAAAABBBBBBCCCCDDDEEF`.

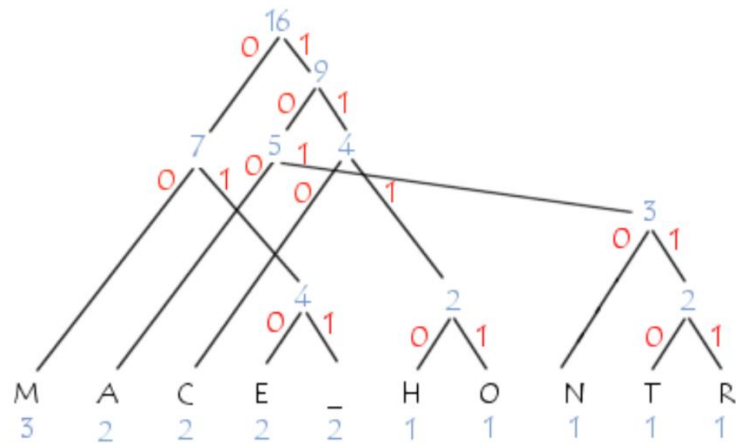
Se usarmos a forma padrão onde o tamanho da representação de cada caractere é fixo, a menor codificação que podemos utilizar para representá-la em binário é de três bits por caractere. Assim temos a seguinte codificação:

Caractere	A	B	C	D	E	F
Código	000	001	010	011	100	101

Para usar o código Huffman e comprimir esta sequência precisamos primeiro montar uma árvore de Huffman seguindo os passos descritos acima. O primeiro passo é contar as ocorrências de cada símbolo na cadeia a ser comprimida. Com isso temos:

Caractere	A	B	C	D	E	F
Contagem	6	5	4	3	2	1

A árvore binária de Huffman é criada, agrupando símbolos por sua frequência, a árvore é percorrida e um dicionário é montado com o código dos símbolos.



Após todo o processo, iniciamos o processo de decodificação na raiz da árvore. Caso o bit seja 0, percorremos a árvore para a esquerda. Caso seja 1, percorremos para a direita. Ao encontrarmos uma folha, imprimimos o símbolo correspondente. Repetimos até não haver mais bits.

HASH

Em ciência da computação, uma tabela de dispersão (também conhecida por tabela de espalhamento ou tabela *hash*, do inglês *hash*) é uma estrutura de dados especial, que associa chaves de pesquisa a valores.

Seu objetivo é, a partir de uma chave simples, fazer uma busca rápida e obter o valor desejado.

A função Hash é qualquer algoritmo que mapeia dados grandes e de tamanho variável para pequenos dados de tamanho fixo. Funções hash aceleram consultas a tabelas ou banco de dados por meio da detecção de registros duplicados em um arquivo grande.

O exemplo abordado em questão foi o da organização dos dados escolares dos alunos nos respectivos escaninhos existentes, os dados são separados de acordo com o número de matrícula de cada um dos alunos. É um problema real, relacionado a organização escolar referente aos alunos, uma vez que possuem um número limitado de escaninhos e uma grande quantidade de alunos, sendo viável essa facilitação para a organização dos dados utilizando os respectivos números de matrícula como é usualmente feito.

No código desenvolvido, criamos um exemplo contendo 10 escaninhos, dando liberdade para o usuário responsável pelo cadastramento adicionar alunos, remover, mostrar e buscar pelos mesmos.

Complexidade de memória	O(1)
Complexidade de tempo	O(n)

BIBLIOGRAFIA

https://pt.wikipedia.org/wiki/Tabela_de_dispers%C3%A3o
<https://br.ccm.net/contents/729-codificacao-de-huffman>
https://pt.wikipedia.org/wiki/Codifica%C3%A7%C3%A3o_de_Huffman
http://www.decom.ufop.br/guilherme/BCC203/2-2019/bcc203_2_2019.htm
<http://www.decom.ufop.br/anascimento/ensino/bcc202/aulas-teoricas/>