UNIVERSIDADE FEDERAL DE OURO PRETO

Ananda Mendes Souza - 19.1.4030

TRABALHO PRÁTICO 3

Ouro Preto - MG
Dezembro 2019

Questão 1: Codificação de Huffman

Em 1950, foi proposto um método estatístico que permitiria atribuir um código binários à vários símbolos a serem comprimidos sendo ele elementos de informação (texto, imagem, áudio, etc) que serão representados por símbolos. A codificação de Huffman é um método de compactação que atribuir códigos menores para símbolos com o uso mais frequente e códigos maiores para símbolos menos usados. Existem diversas forma de compactar dados, uma das mais usada é a busca de reduzir o número de bits que representa os símbolos mais frequentes. Com isso, Huffman constrói uma técnica que tem uma complexidade de tempo excelente - O(n.logn).

Funcionamento:

O processo de codificação é dividido em 4 passo:

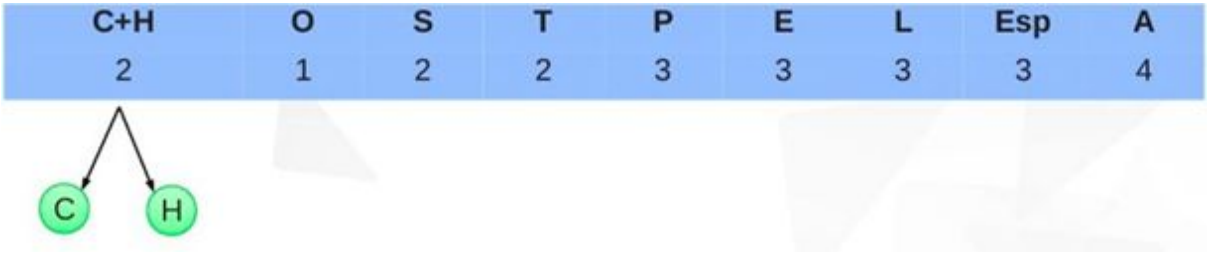
1. Contar a frequência dos símbolos e organizar em ordem de frequência
2. Montar uma árvore binária, agrupando os símbolos pela sua frequência
3. Percorrer a árvore para montar o dicionário com o novo código de cada símbolo;
4. Re-codificar os dados usando o passo 3.

Exemplo: Queremos codificar a mensagem - “CASA_PAPEL_HOTEL_PASTEL”

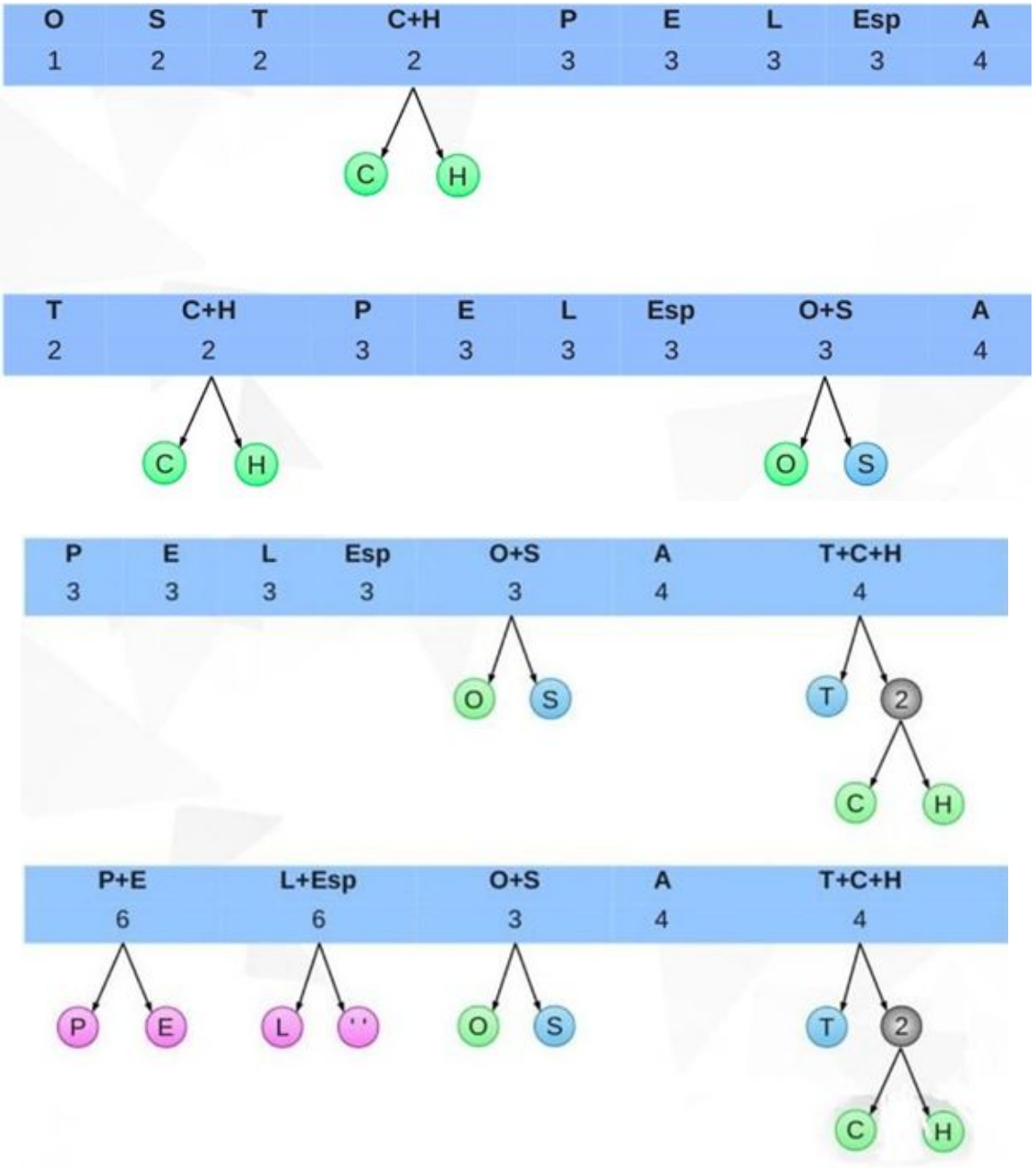
Texto codificado: 0000 0001 00010 0001 1001 0011 0001 0011 0100 0101 1001 0110 0111 1000 0100 0101 1001 0011 0001 0010 1000 0100 0101 - Total de bits: 92

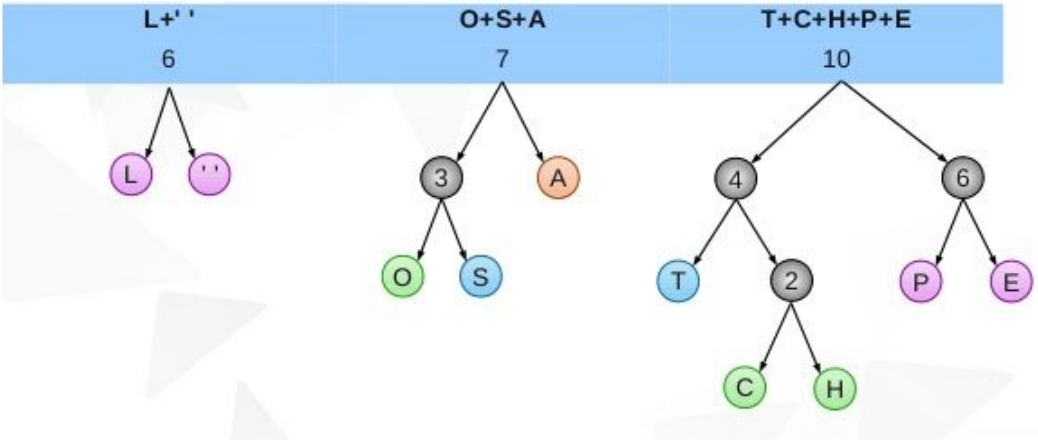
Letra	Frequencia	Letra	Código
C	1	C	0000
A	4	A	0001
S	2	S	0010
P	3	P	0011
E	3	E	0100
L	3	L	0101
H	1	H	0110
O	1	O	0111
T	2	T	1000
Espaço	3	Espaço	1001

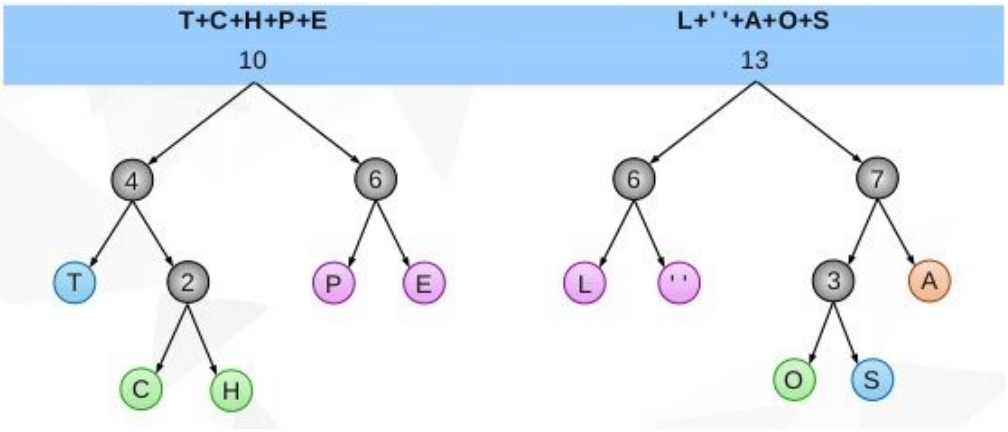
Após de verificar a frequência de letras, iremos construir a árvore binária.

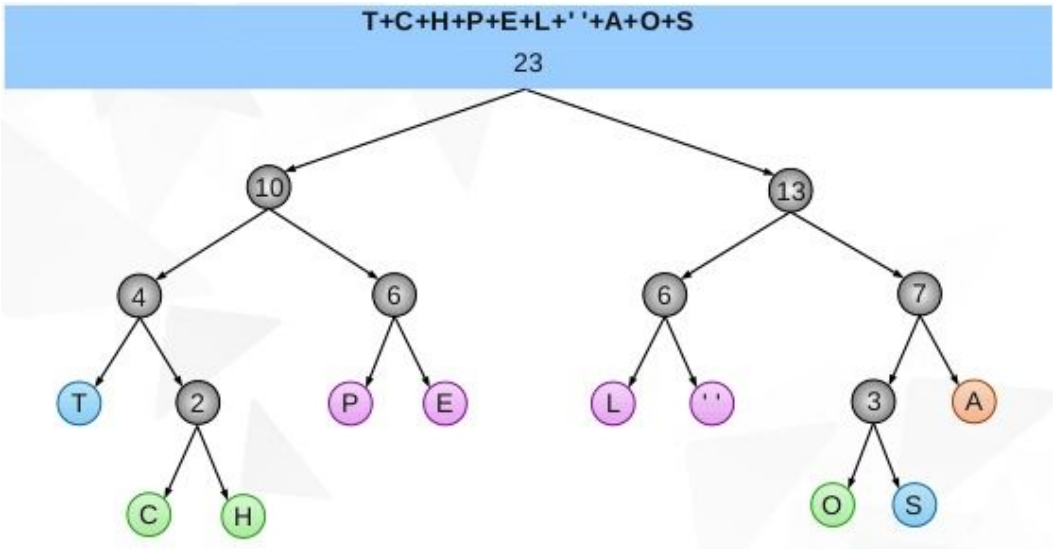


A Figura acima mostra a junção das letras “C” e “H”. A frequência de cada letra é 1.

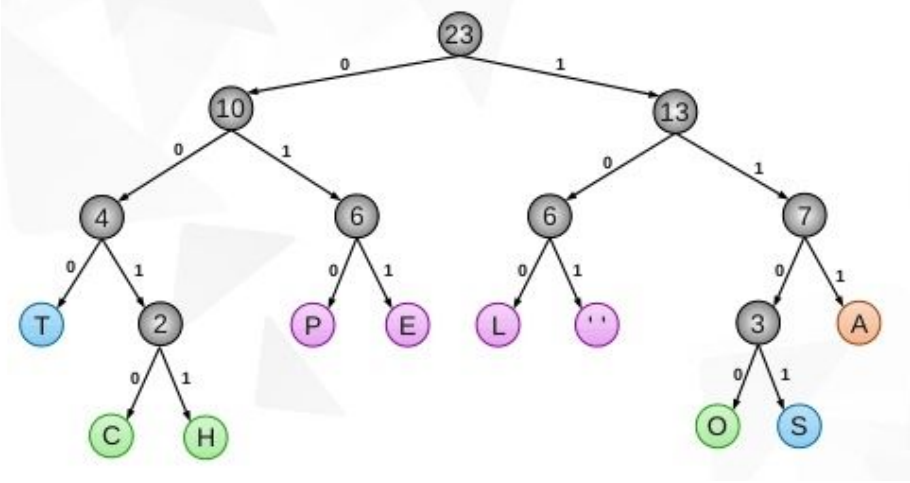



L+' '
6
O+S+A
7
T+C+H+P+E
10
L
' '
3
A
O
S
4
6
T
2
P
E
C
H


T+C+H+P+E
10
L+' '+A+O+S
13
4
6
T
2
P
E
C
H
6
7
L
' '
3
A
O
S


T+C+H+P+E+L+' '+A+O+S
23
10
13
4
6
6
7
T
2
P
E
L
' '
3
A
C
H
O
S

Atribuindo a árvore:



Letra	Código
C	0010
A	111
S	1101
P	010
E	011
L	100
H	0011
O	1100
T	000
Esp.	101

O texto codificado fica assim:

0010 111 1101 111 101 010 111 010 011 100 101 0011 1100 000 011 100 101 010 111 1101 00 011 100 - Total de bit: 74

Dessa forma, a taxa de compressão foi de 20%.

Questão 2: Problema com Tabela Hash

Problema: Uma empresa deseja implementar uma função nova em seu programa principal para agilizar os processos dentro do setor administrativo. Para isso foi feito uma ficha de cadastro de funcionários contendo:

- Nome do funcionário;
- Telefone;
- Cargo na empresa.

Implementação: Foi utilizado tabela hash com lista, para implementar a solução para o problema, uma vez que a empresa teria um acesso rápido de dados de telefone dos funcionários para que facilitasse o contato diretamente, sendo necessário apenas o nome que deseja procurar.

Complexidade: Com a utilização da tabela hash, a as operações de inserção, remoção e pesquisa passam a ter a complexidade de Ω(1) mais a busca na lista encadeada (O(1+N/M)). Porém, a lista terá um tamanho de N/M sendo N o quantidade de registros na tabela e M o

tamanho da tabela, fazendo com que a busca na lista seja menor e diminuindo a complexidade do algoritmo. Caso os valores forem bem distribuídos, evitando colisões, e N for igual a M, a complexidade ficará de $\Omega(1)$. Portanto, a complexidade de tempo final esperada é de $O(1+N/M)$ caso houverem muitas colisões, caso não haja, será de $\Omega(1)$.