

INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA

Licenciatura Engenharia informática e multimédia

Trabalho 2

**Codificação Sinais Multimédia**

**Docentes:**

José Nascimento

**Trabalho realizado por:**

Miguel Távora Nº45102

João Cunha Nº45412

Arman Freitas Nº45414

14/04/2019

INTRODUÇÃO:

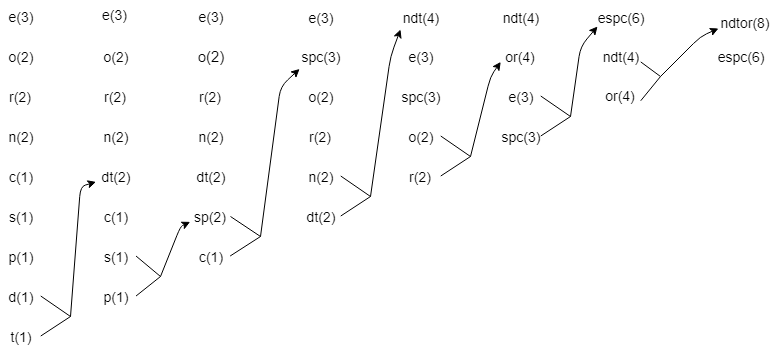
O trabalho foi imposto com o objetivo de aprefeiçoar as habilidades do grupo no que toca á construção de algoritmos. Insidindo neste trabalho na compressão e descompressão de diversos tipos de ficheiros.

Neste trabalho utilizamos a codificação de Huffman, que não possuí perdas e tem uma baixa taxa de redundância. O algoritmo tira por base as probabilidades de cada simbolo na mensagem, contruindo uma árvore agregando sempre os últimos dois caracteres com menor número de ocorrências. Recolocando o novo caracter obtido lista por ordem de probabilidade sucessivamente até so ter dois caracteres com diversos simbolos agregados. Atribuindo-se então para cada aresta da árvore um digito binário(0 ou 1) sendo o código obtido quando é percorrida a árvore e anotando nas arestas os digitos binários desde a raiz até as folhas que corresponde ao simbolo desejado.

Desenvolvimento:

Função gera\_huffman

Para a realização do primeiro era necessário realizar a função que gerava o código de huffman. Para gerar a função gera\_huffman primeiramente foi criada uma função chamada geraDicionario tem como parâmetro o texto introduzido. Apartir do texto introduzido ele gera um dicionario com o número de vezes que o carácter se repete e o carácter correspondente. Após ser gerado o dicionário era indespensável criar uma função que ordenada de forma crescente os caracteres por número de vezes que o mesmo se repete. Para isso foi criada a função ordemCrescente. Na função gera\_huffman propriamente dita gera um código binário correspondente a cada carácter. Para isso foi implementada como função recursiva, onde utilizando uma tabela como argumento, onde vai acrescentando a cada caracter um novo numero obtendo desta forma uma tabela.

Exemplo de um algoritmo de Huffman com a palavra (correspondente):

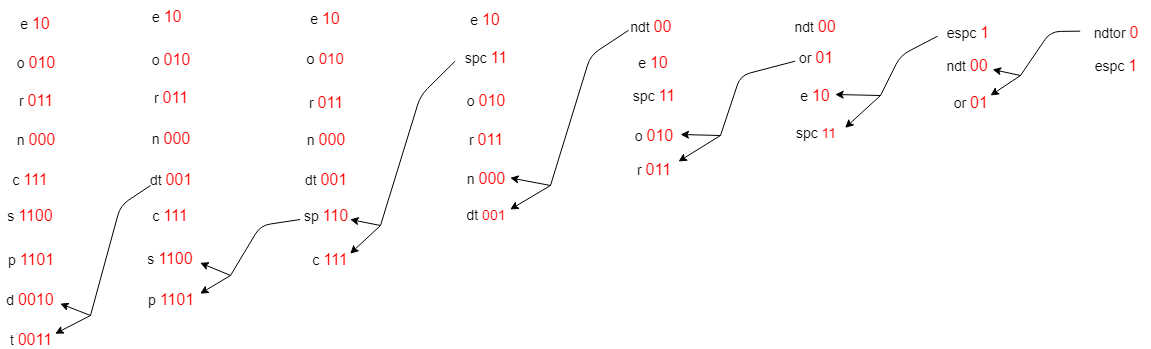
Os resultados difentes resultados obtidos e o tempo que demora cada função para cada executar os diferentes tipos de ficheiros.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | gera\_huffman | codificacao | descodificacao | entropia | nmr medio | eficiencia |
| ecg.txt | 0,31941724 | 0,67318273 | 8,35173678 | 3,26516166 | 9,82352941 | 0,33238173 |
| ubuntu.txt | 2,02020431 | 4,6073401 | 82,2575569 | 4,6779341 | 9,63366337 | 0,48558206 |
| mp3 | 5,94306684 | 26,2587106 | 169,2293 | 4,28631471 | 8,85263158 | 0,48418537 |
| midi | 0,19938469 | 0,93242192 | 10,5513976 | 3,73814176 | 9,37894737 | 0,3985673 |
| imagem |  |  |  |  |  |  |

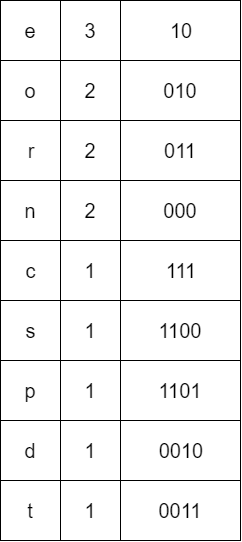
Função Codifica

Para codificar ele recebe uma mensagem (sequência de caracteres) da qual com o código huffman criado anteriormente é possível obter uma sequência de um codigo binário para os caracteres obtidos no codigo de huffman. Atribuindo ao nível mais alto a um caracter o numero 0 e a outro o numero 1, no nível que se segue que faz aresta na árvore atribui-se mais um 0 e ao outro 1 e assim sucessimente para todos os caracteres isolados e obtidos através do codigo de huffman.

Atribuição do código para cada caracter obtendo:



A partir do qual é possível deduzir a tabela correspondente para cada carácter utilizado na mensagem:



Função Descodifica

Para a criação da descodificação é necesário ter a tabela que identifica cada codigo binário e a sequência do código binário obtido pela função de codificação. A partir disso a função descodifica lê os códigos binários e verifica a tabela com os simbolos correspondentes sempre que encontra um caracter conhecido pela tabela ela devolve esse caracter. Executando a ordem sistematicamente até ao fim do código binário. Obtendo desta forma a mensagem codificada.

Exemplo da descodificação da palavra codifica anteriormente(correspondente):



Função Escrever

Tal como o nome diz a função escreve a mensagem a partir do código binário cedido como argumento. Sendo que essa mensagem é guardada em um ficheiro do tipo txt esse mesmo código binário.

Função Ler

Apartir do ficheiro texto criado na função escrever a função ler vai ler a sequência de bits presente dentro do ficheiro txt.

Conclusão

Em suma com a realização deste trabalho o grupo aprendeu a construir tabelas, das quais é possível atribuir valores binários para cada caracter. Permitindo fazer a codificação e descodificação de mensagens. Conseguindo desta forma optimizer bastante todo o tipo de ficheiro, sendo um dos melhores métodos para codificar a mensagem a codificação de Huffman. A codificação permite algoritmos ótimis em termos de eficiência e não possuí perdas, mas a caracteristica que o torna distinto dos outros é o facto de ser um algoritmo não redundante, ou seja, uma sequência de números binários nunca pode dar 2 caracteres diferentes.