Ana Rita Almeida, Pedro Sousa

UFP | 2021-2022

Método Run Length Encoding (RLE)

Multimédia II

Índice

[Resumo 2](#_Toc103441234)

[1. Introdução 3](#_Toc103441235)

[1.1. Objetivos do trabalho 3](#_Toc103441236)

[1.2. Justificação 3](#_Toc103441237)

[1.3. Áreas de aplicação 3](#_Toc103441238)

[2. O algoritmo Run Length Encoding 4](#_Toc103441239)

[3. Implementação do algoritmo Run Length Encoding 5](#_Toc103441240)

[3.1. Compressão de texto 5](#_Toc103441241)

[3.2. Descompressão de texto 6](#_Toc103441242)

[3.3. Compressão de imagem 7](#_Toc103441243)

[3.4. Descompressão de imagem 8](#_Toc103441244)

[4. Comparação com outros métodos 9](#_Toc103441245)

[5. Conclusão 10](#_Toc103441246)

[Bibliografia 11](#_Toc103441247)

# Resumo

Este artigo científico foi desenvolvido para a disciplina de Multimédia II e consiste em estudar o método Run Length Encoding e explorar os seus algoritmos para codificação e descodificação de texto e imagens.

# Introdução

O Run Length Encoding é um processo de compressão de caracteres quando existe uma longa sequência dos mesmos (4 ou mais). É um método considerado simples e o seu codec (codificador/descodificador) são sem perdas e supressão de sequências repetitivas.

## Objetivos do trabalho

## 

(A RETIFICAR) O objetivo deste trabalho é desenvolvermos uma aplicação de software que implemente um codec multimédia. O trabalho será maioritariamente realizado em javascript, com HTML e CSS para ficar disponível em versão Web.

## Justificação

Entre os elementos do grupo, achamos que o RLE seria uma boa opção. Embora este método não seja dos mais eficientes (devido às restrições dos números de repetições de caracteres), é um algoritmo muito utilizado e também cumpre o requisito 1 que impõe a implementação de um algoritmo de compressão/descompressão **sem perdas**.

## Áreas de aplicação

Este codec é usado para a compressão de texto e imagens, especialmente em áreas onde não se pode perder informação pois estes métodos são sem perdas.

# O algoritmo Run Length Encoding

Run Length Encoding é um método de compressão sem perdas de dados onde uma determinada sequencia de valores repetidos são combinados como um único valor e a quantidade de valores repetidos no lugar da sequência original.

Este algoritmo é especialmente útil e eficaz para grandes conjuntos de repetições.

# Implementação do algoritmo Run Length Encoding

## Compressão de texto

O processo de codificação por RLE é o seguinte:

* Primeiro determinamos uma Flag que não exista no texto que vamos comprimir, como um caractere especial por exemplo.
* Em seguida vamos ler o caractere na posição i (primeira posição, a inicial). Se o caractere seguinte (i+1) não for igual ao da posição i (texto[i+1] != texto[i]), i+1 passa a ser o novo i e começamos novamente a sequência. Se i+1 for igual a i, vamos verificando os caracteres seguintes (i+2, i+3, etc.) enquanto os caracteres nessas posições sejam iguais ao caractere na posição i (texto[i]).
* Quando essa sequência acabar vamos contar o número de vezes que o caractere se repetiu.
* Se a sequência for menor que 4, não se efetua compressão e então os caracteres vão permanecer iguais.
* Se a sequência de caracteres repetidos for igual ou maior que 4, faz-se a compressão e o resultado final será algo como: Flag + nº de repetições + caracter.

Exemplo:

Vamos selecionar a Flag “$”

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| A | A | A | A | B | B | B | C | C | C | C | C | D | D | D | D | D | D |

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| $ | 4 | A | B | B | B | $ | 5 | C | $ | 6 | D |

## Descompressão de texto

O processo de descodificação por RLE é o seguinte:

* Primeiro é determinada a Flag que está a ser usada.
* Criamos uma string auxiliar vazia.
* Começamos a ler o nosso texto até encontrarmos a Flag.
* Quando encontramos a Flag, estamos na posição texto[i]. Na posição

texto[i + 1] vamos encontrar o numero de vezes que um caractere vai surgir,

em texto[i + 2] vamos encontrar o caractere repetido.

* Em seguida, criamos uma string de repetições com o tamanho do número de repetições do caractere e colocamos esse caractere em todas as posições.
* Vamos fazer a concatenação da String auxiliar que criamos no início com a string de repetições. E depois zeramos a string de repetições para voltar a usar porque cada vez que aparecer a Flag na string texto, vamos repetir o ponto anterior, seguido deste ponto.

Exemplo:

Flag: #

i i+1 i+2

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| # | 5 | A | B | B | # | 6 | C |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| A | A | A | A | A |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| B | B | # | 6 | C |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| A | A | A | A | A | B | B |

|  |  |  |
| --- | --- | --- |
| # | 6 | C |

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| A | A | A | A | A | B | B | C | C | C | C | C | C |

## Compressão de imagem

## Descompressão de imagem

# Comparação com outros métodos

# Conclusão

# Bibliografia

(corrigir!)

<http://multimedia.ufp.pt/codecs/compressao-sem-perdas/supressao-de-sequencias-repetitivas/run-length-encoding/>