

# COMPRESSÃO DE DADOS MÉTODOS DE DICIONÁRIO BURROWS WHEELER

---

Sérgio Mergen

# LZW (Lempel-Ziv-Welch)

- Método de compressão LZW
  - Extensão do LZ78
  - Usa códigos simples em vez de duplas
- Ideia:
  - Encontrar o maior prefixo S no dicionário
  - Não codificar o próximo caractere c
    - Mas adicionar Sc ao dicionário

# LZW (Lempel-Ziv-Welch)

- Dicionário
  - Inicializado com n entradas preenchidas
  - n = tamanho do alfabeto
- Ex. Considerando a tabela ASCII como alfabeto
  - O dicionário viria com 256 entradas iniciais

Decimal	Binário	Caractere
0	00000000	
...		...
97	01100001	a
98	01100010	b
99	01100011	c
...		...
255	11111111	

# Exemplo de codificação – LZW

a	a	b	a	a	c	a	b
---	---	---	---	---	---	---	---

## Dicionário

pos	padrão
-----	--------

# Exemplo de codificação – LZW

A maior sequência no dicionário é (a), na posição 97.

Adicionar código com pos = 97

Adicionar no dicionário o caractere (a)

+ o próximo caractere (a)

a	a	b	a	a	c	a	b
---	---	---	---	---	---	---	---

 (97)

## Dicionário

pos	padrão
256	aa

# Exemplo de codificação – LZW

A maior sequência no dicionário é (a), na posição 97

Adicionar código com pos = 97

Adicionar no dicionário o caractere (a)

+ o próximo caractere (b)

a	a	b	a	a	c	a	b
---	---	---	---	---	---	---	---

 (97)

a	a	b	a	a	c	a	b
---	---	---	---	---	---	---	---

 (97)

## Dicionário

pos	padrão
256	aa
257	ab

# Exemplo de codificação – LZW

A maior sequência no dicionário é (b), na posição 98.

Adicionar código com pos = 98

Adicionar no dicionário a sequência (b)  
+ o próximo caractere (a)

a	a	b	a	a	c	a	b
---	---	---	---	---	---	---	---

 (97)

a	a	b	a	a	c	a	b
---	---	---	---	---	---	---	---

 (97)

a	a	b	a	a	c	a	b
---	---	---	---	---	---	---	---

 (98)

## Dicionário

pos	padrão
256	aa
257	ab
258	ba

# Exemplo de codificação – LZW

A maior sequência no dicionário é (aa), na posição 256.

Adicionar código com pos = 256

Adicionar no dicionário a sequência (aa)  
+ o próximo caractere (c)

a	a	b	a	a	c	a	b	(97)
a	a	b	a	a	c	a	b	(97)
a	a	b	a	a	c	a	b	(98)
a	a	b	a	a	c	a	b	(256)

## Dicionário

pos	padrão
256	aa
257	ab
258	ba
259	aac



# Exemplo de codificação – LZW

A maior sequência no dicionário é (c), na posição 99.

Adicionar código com pos = 99

Adicionar no dicionário o caractere (c)

+ o próximo caractere (a)

a	a	b	a	a	c	a	b
---	---	---	---	---	---	---	---

 (97)

a	a	b	a	a	c	a	b
---	---	---	---	---	---	---	---

 (97)

a	a	b	a	a	c	a	b
---	---	---	---	---	---	---	---

 (98)

a	a	b	a	a	c	a	b
---	---	---	---	---	---	---	---

 (256)

a	a	b	a	a	c	a	b
---	---	---	---	---	---	---	---

 (99)

## Dicionário

pos	padrão
256	aa
257	ab
258	ba
259	aac
260	ca

# Exemplo de codificação – LZW

A maior sequência no dicionário é (ab), na posição 257.

Adicionar código com pos = 257

Fim

a	a	b	a	a	c	a	b	(97)
a	a	b	a	a	c	a	b	(97)
a	a	b	a	a	c	a	b	(98)
a	a	b	a	a	c	a	b	(256)
a	a	b	a	a	c	a	b	(99)
a	a	b	a	a	c	a	b	(257)

## Dicionário

pos	padrão
256	aa
257	ab
258	ba
259	aac
260	ca

# Exemplo de decodificação – LZW

(97)	<table><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr></table>								
(97)	<table><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr></table>								
(98)	<table><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr></table>								
(256)	<table><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr></table>								
(99)	<table><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr></table>								
(257)	<table><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr></table>								

Dicionário

pos	padrão
-----	--------

# Exemplo de decodificação – LZW

A entrada 97 do dicionário é copiada para a saída

(97)	a						
(97)							
(98)							
(256)							
(99)							
(257)							

**Dicionário**

pos	padrão
-----	--------

# Exemplo de decodificação – LZW

A entrada 97 do dicionário é copiada para a saída

O código anterior (97= a) +

o primeiro caractere do código atual (97= a) são copiados para o dicionário

(97)	a						
(97)	a	a					
(98)							
(256)							
(99)							
(257)							

## Dicionário

pos	padrão
256	aa

# Exemplo de decodificação – LZW

A entrada 98 do dicionário é copiada para a saída

O código anterior (97= a) +

o primeiro caractere do código atual (98= b) são copiados para o dicionário

(97)	a						
(97)	a	a					
(98)	a	a	b				
(256)							
(99)							
(257)							

Dicionário

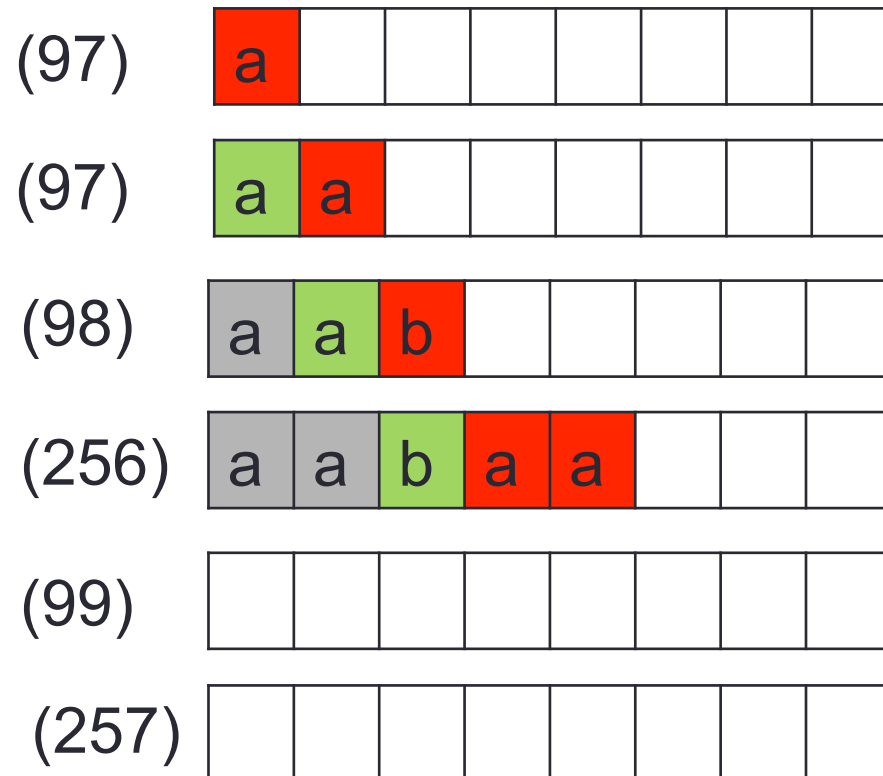
pos	padrão
256	aa
257	ab

# Exemplo de decodificação – LZW

A entrada 256 do dicionário é copiada para a saída

O código anterior (98= b) +

o primeiro caractere do código atual (256 = aa) são copiados para o dicionário



**Dicionário**

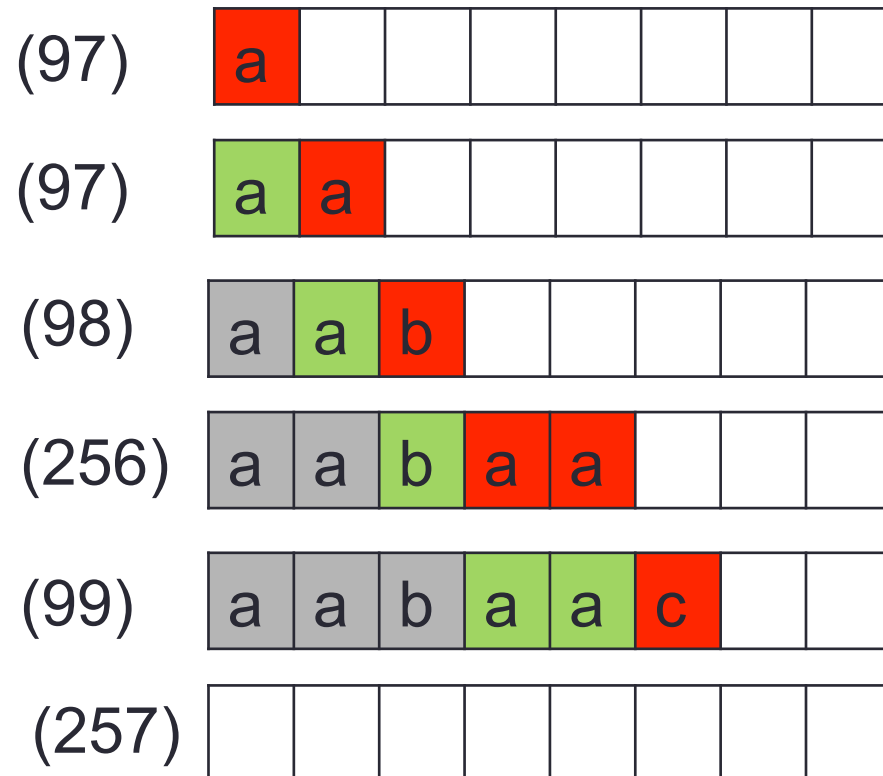
pos	padrão
256	aa
257	ab
258	ba

# Exemplo de decodificação – LZW

A entrada 99 do dicionário é copiada para a saída

O código anterior (256= aa) +

o primeiro caractere do código atual (99= c) são copiados para o dicionário



**Dicionário**

pos	padrão
256	aa
257	ab
258	ba
259	aac



# Exemplo de decodificação – LZW

A entrada 257 do dicionário é copiada para a saída

O código anterior (99= c) +

o primeiro caractere do código atual (257 = a) são copiados para o dicionário

(97)	a						
(97)	a	a					
(98)	a	a	b				
(256)	a	a	b	a	a		
(99)	a	a	b	a	a	c	
(257)	a	a	b	a	a	c	a b

**Dicionário**

pos	padrão
256	aa
257	ab
258	ba
259	aac
260	ca

# Análise

- Todos os métodos de dicionário vistos
  - Tem taxas de compressão semelhantes
- LZ77
  - usa menos memória
- LZ78 / LZT
  - mais eficientes
  - Número de comparações de cadeias durante codificação é menor
    - Graças ao uso da Trie

# Análise

- Problemas das abordagens LZ78 e LZW
  - O dicionário pode crescer demais
- Solução 1: Estagnar o dicionário
  - Quando ele atingir um tamanho determinado
  - Quando a razão de compressão não mudar mais muito
- Vantagens:
  - Taxa de compressão não apresenta perdas consideráveis
  - Custo de memória não ultrapassa um limiar

# Análise

- Problemas das abordagens LZ78 e LZW
  - O dicionário pode crescer demais
- Solução 2: Descartar entradas do dicionário
  - Aquelas menor recentemente usadas (LRU)
  - Quando o dicionário atingir um tamanho determinado
- Aplicações:
  - Usado no padrão BLTZ, da British Telecom

# Análise

- Problemas das abordagens LZ78 e LZW
  - O dicionário pode crescer demais
- Solução 3: Descartar o dicionário antigo e criar um dicionário novo
  - Quando ele atingir um determinado tamanho
    - Usado no GIF
  - Quando ele não for mais eficiente na compressão
    - Usado pelo *compress*, do *unix*
      - LZC (Lempel-Ziv-Compress)

# Análise

- Problemas das abordagens LZ78 e LZW
  - O dicionário pode crescer demais
- Solução 3: Descartar o dicionário antigo e criar um dicionário novo
- *Vantagens:*
  - Adapta-se bem a mudanças no padrão de informações a compactar.
  - Ex. quando compacta-se arquivos diferentes em um só arquivo

# Análise

- Versões iniciais não usavam probabilidade de ocorrência dos caracteres
  - e obtinham uma taxa de compressão mais baixa
- Versões mais recentes aliam dicionários (ex. LZ77) e métodos estatísticos (ex. Hufmann)
  - e obtém taxas de compressão mais altas
  - Ex. GZIP
    - Método de compressão DEFLATE

# Análise

- De modo geral
  - LZ77 é melhor, mas mais lenta
  - mas a versão gzip do LZ77 é quase tão rápida quanto qualquer LZ78.
- Versões comerciais não são implementações puras das estratégias apresentadas
  - Mas são fortemente baseadas nelas



# Burrows Wheeler

- O método de Burrows Wheeler busca transformar uma sequência de caracteres em uma sequência mais adequada para codificação
- A transformação normalmente não gera compressão
  - Mas a sequência transformada é mais propícia para compressão
- O método se baseia no fato de que alguns caracteres costumam aparecer próximos em palavras
  - Marco
  - Parto
  - Cargo

# Burrows Wheeler

- A compressão usando o método de Burrows Wheeler passa por três etapas
  - BWT – Transformação de Burrows Wheelers
    - Transforma a cadeia original em outra mais adequada à compressão
  - MTF – Move to Front
    - Altera a frequência dos símbolos
    - Fazendo com que poucos símbolos sejam muito frequentes
  - Huffman – Codificação de Huffman
    - Aqui é onde a compressão realmente acontece



# Burrows Wheeler

- A compressão usando o método de Burrows Wheeler passa por três etapas
  - BWT – Transformação de Burrows Wheelers
    - Transforma a cadeia original em outra mais adequada à compressão
  - MTF – Move to Front
    - Altera a frequência dos símbolos
    - Fazendo com que poucos símbolos sejam muito frequentes
  - Huffman – Codificação de Huffman
    - Aqui é onde a compressão realmente acontece

BOMBABAMBA



# BWT

- Passo 1: Permutar
  - Criar uma matriz  $n \times n$
  - Preencher a matriz com permutações dos caracteres do texto
  - Na linha  $i$ 
    - Fazer uma rotação de uma posição
      - do texto da linha  $i-1$
- Passo 2: Ordenar
  - Ordenar as permutações
- Passo 3: Transformar
  - Criar o código de saída
  - Usando a última coluna da matriz















# Exemplo - BWT

- Passo 1: Permutar

[illegible]





# Exemplo - BWT

- Passo 1: Permutar

1	B	O	M	B	A	B	A	M	B	A
2	A	B	O	M	B	A	B	A	M	B
3	B	A	B	O	M	B	A	B	A	M
4	M	B	A	B	O	M	B	A	B	A
5	A	M	B	A	B	O	M	B	A	B
6	B	A	M	B	A	B	O	M	B	A
7	A	B	A	M	B	A	B	O	M	B
8	B	A	B	A	M	B	A	B	O	M
9	M	B	A	B	A	M	B	A	B	O
10	O	M	B	A	B	A	M	B	A	B

# Exemplo - BWT

- Passo 2: Ordenar

1	B	O	M	B	A	B	A	M	B	A
2	A	B	O	M	B	A	B	A	M	B
3	B	A	B	O	M	B	A	B	A	M
4	M	B	A	B	O	M	B	A	B	A
5	A	M	B	A	B	O	M	B	A	B
6	B	A	M	B	A	B	O	M	B	A
7	A	B	A	M	B	A	B	O	M	B
8	B	A	B	A	M	B	A	B	O	M
9	M	B	A	B	A	M	B	A	B	O
10	O	M	B	A	B	A	M	B	A	B

# Exemplo - BWT

- Passo 2: Ordenar

1	A	B	A	M	B	A	B	O	M	B
2	A	B	O	M	B	A	B	A	M	B
3	A	M	B	A	B	O	M	B	A	B
4	B	A	B	A	M	B	A	B	O	M
5	B	A	B	O	M	B	A	B	A	M
6	B	A	M	B	A	B	O	M	B	A
7	B	O	M	B	A	B	A	M	B	A
8	M	B	A	B	A	M	B	A	B	O
9	M	B	A	B	O	M	B	A	B	A
10	O	M	B	A	B	A	M	B	A	B



# Exemplo - BWT

- Passo 3: Transformar



1	A	B	A	M	B	A	B	O	M	B
2	A	B	O	M	B	A	B	A	M	B
3	A	M	B	A	B	O	M	B	A	B
4	B	A	B	A	M	B	A	B	O	M
5	B	A	B	O	M	B	A	B	A	M
6	B	A	M	B	A	B	O	M	B	A
7	B	O	M	B	A	B	A	M	B	A
8	M	B	A	B	A	M	B	A	B	O
9	M	B	A	B	O	M	B	A	B	A
10	O	M	B	A	B	A	M	B	A	B



Saída: 7 BBBMMAAOAB

# Burrows Wheeler

- A transformação de Burrows Wheeler é reversível
- Pode-se obter o texto original
- A partir dos dados codificados
  - o índice da linha do texto original
  - Os caracteres da última coluna
- Ex.
  - Entrada: 7 BBBMMAAOAB
  - Saída: **BOMBABAMBA**

# Burrows Wheeler

- A compressão usando o método de Burrows Wheeler passa por três etapas
  - BWT – Transformação de Burrows Wheelers
    - Transforma a cadeia original em outra mais adequada à compressão
  - MTF – Move to Front
    - Altera a frequência dos símbolos
    - Fazendo com que poucos símbolos sejam muito frequentes
  - Huffman – Codificação de Huffman
    - Aqui é onde a compressão realmente acontece



# Move to Front

- O MTF gera sequências melhores para a etapa seguinte
  - Frequências altas de números baixos
  - Muitos zeros
- O código de huffman tira proveito dessas sequências
  - Poucos símbolos com alta frequência
  - Muitos símbolos com baixa frequência
- Exemplo:
  - 1-0-0-2-0-1-0-0-0-6-0-1-0-3-0-1-0-0-1-1

# Move to Front

- Passo 1
  - Criar uma tabela de códigos para todos os símbolos possíveis
    - Normalmente são 256 códigos de 8 bits cada
- Passo 2
  - Para cada caractere na entrada
    - Encontrar código correspondente
    - Usar código na saída
    - Mover código para a primeira posição da tabela
    - Deslocar demais códigos para baixo

# Exemplo - MTF

- Sem usar MTF
- Entrada:
  - BBBMMAAOAB
- Saída:
  - 1112200301

A	0
B	1
M	2
O	3

Tabela de códigos

# Exemplo - MTF

- Usando MTF
- Entrada:
  - **B**BBMMAAOAB
- Saída:
  - **1**

antes			depois	
A	0	➔	B	0
B	1		A	1
M	2		M	2
O	3		O	3

Tabelas de códigos

# Exemplo - MTF

- Usando MTF
- Entrada:
  - B<sup>B</sup>BMMAAOAB
- Saída:
  - 1<sup>0</sup>

antes			depois	
B	0	➔	B	0
A	1		A	1
M	2		M	2
O	3		O	3

Tabelas de códigos



# Exemplo - MTF

- Usando MTF
- Entrada:
  - BB**B**MMAAOAB
- Saída:
  - 10**0**

antes			depois	
B	0	➔	B	0
A	1		A	1
M	2		M	2
O	3		O	3

Tabelas de códigos

# Exemplo - MTF


- Usando MTF
- Entrada:
  - BBBMMAAOAB
- Saída:
  - 1002

antes			depois	
B	0	➔	M	0
A	1		B	1
M	2		A	2
O	3		O	3

Tabelas de códigos

# Exemplo - MTF

- Usando MTF
- Entrada:
  - BBBMMAAOAB
- Saída:
  - 10020

antes			depois	
M	0		M	0
B	1		B	1
A	2		A	2
O	3		O	3

Tabelas de códigos

# Exemplo - MTF

- Usando MTF
- Entrada:
  - BBBMM~~A~~AOAB
- Saída:
  - 10020~~2~~

antes			depois	
M	0	➔	A	0
B	1		M	1
A	2		B	2
O	3		O	3

Tabelas de códigos

# Exemplo - MTF

- Usando MTF
- Entrada:
  - BBBMMA<sup>A</sup>OAB
- Saída:
  - 100202<sup>0</sup>

antes			depois	
A	0	➔	A	0
M	1		M	1
B	2		B	2
O	3		O	3

Tabelas de códigos

# Exemplo - MTF

- Usando MTF
- Entrada:
  - BBBMMAA**O**AB
- Saída:
  - 1002020**3**

antes			depois	
A	0	➔	O	0
M	1		A	1
B	2		M	2
O	3		B	3

Tabelas de códigos

# Exemplo - MTF

- Usando MTF
- Entrada:
  - BBBMMAOAB
- Saída:
  - 100202031

antes			depois	
O	0	➔	A	0
A	1		O	1
M	2		M	2
B	3		B	3

Tabelas de códigos

# Exemplo - MTF

- Usando MTF
- Entrada:
  - BBBMMAAOAB
- Saída:
  - 100202031

antes			depois	
A	0	➔	B	0
O	1		A	1
M	2		O	2
B	3		M	3

Tabelas de códigos



# Burrows Wheeler

- A compressão usando o método de Burrows Wheeler passa por três etapas
  - BWT – Transformação de Burrows Wheelers
    - Transforma a cadeia original em outra mais adequada à compressão
  - MTF – Move to Front
    - Altera a frequência dos símbolos
    - Fazendo com que poucos símbolos sejam muito frequentes
  - Huffman – Codificação de Huffman
    - Aqui é onde a compressão realmente acontece



# Código de Huffman

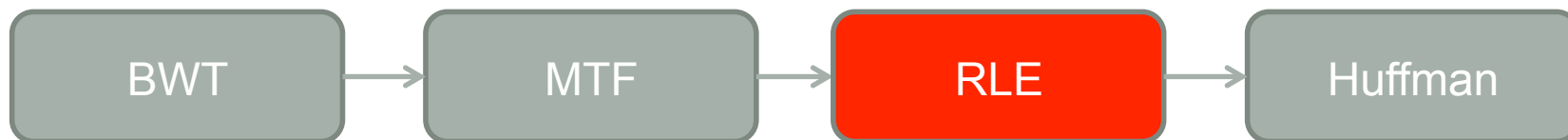
- Monta uma tabela com a frequência de cada símbolo
- Usar a frequência para criar a árvore de Huffman
  - Conforme visto em aula
- O código obtém boa compressão quando
  - Poucos símbolos são muito frequentes
  - Esses símbolos são codificados com poucos bits

# Exemplo - Huffman

- Comparação das tabelas de frequência usadas pelo código de Huffman
- Sem usar MTF
  - 1112200301
    - 0 = 3
    - 1 = 4
    - 2 = 2
    - 3 = 1
- Usando MTF
  - 1002020313
    - 0 = 4
    - 1 = 2
    - 2 = 2
    - 3 = 2
- Nesse caso, o MTF não ajudou
  - Para textos maiores, a frequência dos símbolos de baixo valor aumenta bastante

# Burrows Wheeler e RLE

- A etapa de MTF pode gerar códigos onde o zero apareça demais em algumas partes
  - E muito pouco em outras partes
- Isso faz com que sua frequência seja a mais alta de todas
  - Mesmo em arquivos onde sua frequência só é alta em determinadas regiões
- Para diminuir essa elevada frequência local de zeros
  - Pode-se usar a codificação de RLE
    - E representar longas repetições de zeros por códigos menores



# Análise

- De modo geral
  - A taxa de compressão é maior do que os compressores baseados em dicionários
  - Porém, são mais lentos (tanto compressão quanto descompressão)
- Quais operações são lentas?
  - As permutações da mensagem
  - E principalmente, a sua ordenação
- É possível otimizar
  - Mas ainda assim, a execução é lenta
- Uso comercial limitado
  - Bzip2 utiliza BWT

# Alguns formatos de compressão existentes

Método	Formato
LZ77	Zip, Pkzip, gzip, png
LZSS	Rar
LZW	Compress, gif
BWT	bzip2

# Exercício

- Dada a seguinte cadeia de caracteres:
  - a\_aba\_da\_aba\_da\_baba
- Mostre a codificação que seria obtida pelos seguintes métodos:
  - LZW
  - BWT

**Tabela ASCII**

Binário	Decimal	Caractere
01011111	95	_
01100001	97	a
01100010	98	b
01100100	100	d