

# MATA54 - Estruturas de Dados e Algoritmos II

## Compressão

Flávio Assis

Versão gerada a partir de slides do Prof. George Lima

IC - Instituto de Computação

Salvador, novembro de 2021

# Motivação: remoção da redundância na representação de informação

## Aplicações

Qualquer aplicação que objetiva a economia de memória (armazenamento) e tempo (transmissão) de dados

## Abordagens a serem estudadas: compressão sem perdas

- ▶ Codificação de Huffman
- ▶ Algoritmo de Lempel-Ziv

# Código e Codificação

Seja a sequência:

**aaaabbcaaaddbcbdeaaaabbbbaa**

Suponha a seguinte representação (hipotética) de 8 bits:

a: 01010001    b: 01110110    c: 10011101    d: 01010101    e: 01100011

A codificação da sequência seria:

<u>01010001</u>	<u>01010001</u>	<u>01010001</u>	<u>01010001</u>	<u>01110110</u>	<u>01110110</u>	<u>10011101</u>
a	a	a	a	b	b	c
<u>01010001</u>	<u>01010001</u>	<u>01010001</u>	<u>01010101</u>	<u>01010101</u>	<u>01110110</u>	<u>10011101</u>
a	a	a	d	d	b	c
<u>01010101</u>	<u>01010101</u>	<u>01100011</u>	<u>01010001</u>	<u>01010001</u>	<u>01010001</u>	<u>01110110</u>
d	d	e	a	a	a	b
<u>01110110</u>	<u>01110110</u>	<u>01110110</u>	<u>01010001</u>	<u>01010001</u>		
b	b	b	a	a		

**208 bits!** É possível representar os dados com menos bits?

# Código e Codificação

Seja a sequência:

**aaaabbcaaaddbcdddeaaabbbbbaa**

Poderíamos considerar a frequência de ocorrência dos caracteres:

a: 12 vezes    b: 7 vezes    c: 2 vezes    d: 4 vezes    e: 1 vez

Caracteres mais frequentes poderiam ter representações menores.

Exemplo:

a: 0    b: 1    c: 00    d: 01    e: 010

A codificação da sequência seria:

0 0 0 0 1 1 00 0 0 0 01 01 1 00 01 01  
a a a a b b c a a a d d b c d d  
010 0 0 0 1 1 1 1 0 0  
e a a a b b b b a a

**34 bits!** Mas há um problema!

# Código e Codificação

Seja a sequência:

**aaaabbcaaaddbcdeaaabbbbbaa**

Poderíamos considerar a frequência de ocorrência dos caracteres:

a: 12 vezes    b: 7 vezes    c: 2 vezes    d: 4 vezes    e: 1 vez

Caracteres mais frequentes poderiam ter representações menores.

Exemplo:

a: 0    b: 1    c: 00    d: 01    e: 010

A codificação da sequência seria:

$$\begin{array}{cccccccccccccccc} \underbrace{0}_a & \underbrace{0}_a & \underbrace{0}_a & \underbrace{0}_a & \underbrace{1}_b & \underbrace{1}_b & \underbrace{00}_c & \underbrace{0}_a & \underbrace{0}_a & \underbrace{0}_a & \underbrace{01}_d & \underbrace{01}_d & \underbrace{1}_b & \underbrace{00}_c & \underbrace{01}_d & \underbrace{01}_d \\ \underbrace{010}_e & \underbrace{0}_a & \underbrace{0}_a & \underbrace{0}_a & \underbrace{1}_b & \underbrace{1}_b & \underbrace{1}_b & \underbrace{1}_b & \underbrace{0}_a & \underbrace{0}_a & & & & & & \end{array}$$

**34 bits!** Mas há um problema! **Redundância da decodificação**

O que fazer? **Nenhuma representação pode ser prefixo de outra!**

# Código e Codificação

Seja a sequência:

**aaaabbcaaaaddbcbddeaaaabbbbbaa**

Codificação livre de prefixos:

a: 1    b: 01    c: 001    d: 0000    e: 0001

A sequência ficaria:

$$\begin{array}{cccccccccccccccc} \underbrace{1}_a & \underbrace{1}_a & \underbrace{1}_a & \underbrace{1}_a & \underbrace{01}_b & \underbrace{01}_b & \underbrace{001}_c & \underbrace{1}_a & \underbrace{1}_a & \underbrace{1}_a & \underbrace{0000}_d & \underbrace{0000}_d & \underbrace{01}_b & \underbrace{001}_c & \underbrace{0000}_d \\ 0000 & 0001 & 1 & 1 & 1 & 01 & 01 & 01 & 01 & 1 & 1 \\ \underbrace{d} & \underbrace{e} & \underbrace{a} & \underbrace{a} & \underbrace{a} & \underbrace{b} & \underbrace{b} & \underbrace{b} & \underbrace{b} & \underbrace{a} & \underbrace{a} \end{array}$$

**52 bits!** Livre de redundância na decodificação!

Existe codificação livre de prefixos melhor que essa para esta sequência?

# Código e Codificação

Atribuição de sequência de bits a símbolos ou sequência de símbolos.

- ▶ Mínima redundância (**desejável**)
- ▶ Tamanho fixo ou variável (variável é **preferível**)
- ▶ Unicamente decodificável (**mandatório**)
- ▶ Decodificação eficiente (**muito importante**)

Ex.: Mensagem: *aaabbcbbaaabbcc*

- ▶  $a : 1; b : 00; c : 01$
- ▶  $a : 0; b : 11; c : 10$
- ▶  $aaa : 0; bb : 11; c : 10$

# Codificação de Huffman



# Huffman (estático): codificação eficiente por símbolo

## Código de prefixo e otimalidade

- ▶ Nenhum código atribuído a um símbolo é prefixo de outros
- ▶ Símbolos que ocorrem com menor frequência possuem maior código que aqueles que ocorrem com maior frequência
- ▶ Código de Huffman (estático) é um código de prefixo ótimo

## Código de Huffman (1952)

Ideia básica: gerar uma árvore binária ponderada

1. Seja  $L$  uma lista dos símbolos ordenada por frequência
2. Retire os dois símbolos  $s_i$  e  $s_j$  de menor frequência  $f_i$  e  $f_j$
3. Gere um novo símbolo  $s$  com frequência  $f = f_i + f_j$  e inclua-o em  $L$
4. Gere um nó numa árvore binária que representa  $s$  com filhos  $s_i$  e  $s_j$
5. Retorne a (2) até que só haja um símbolo (raíz da árvore)
6. Gere um código para cada símbolo  $s_i$  representando o caminho da raiz até  $s_i$

# Huffman: Ilustração

## Codificação de Huffman

Sequência: **abbdacccaabadaeeaa**

Frequências:

**a**: 8/18    **b**: 3/18    **c**: 3/18    **d**: 2/18    **e**: 2/18

Menores frequências: **e** e **d**

Nova lista:

**a**: 8/18    **b**: 3/18    **c**: 3/18    **de**: 4/18



# Huffman: Ilustração

## Codificação de Huffman

Sequência: **abbdacccaabadaeeaa**

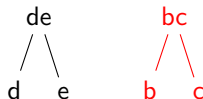
Lista:

**a**: 8/18    **b**: 3/18    **c**: 3/18    **de**: 4/18

Menores frequências: **b** e **c**

Nova lista:

**a**: 8/18    **bc**: 6/18    **de**: 4/18



# Huffman: Ilustração

## Codificação de Huffman

Sequência: **abbdacccaabadaeeaa**

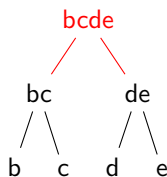
Lista:

**a**: 8/18    **bc**: 6/18    **de**: 4/18

Menores frequências: **bc** e **de**

Nova lista:

**a**: 8/18    **bcde**: 10/18



# Huffman: Ilustração

## Codificação de Huffman

Sequência: **abbdaccaabadaeeaa**

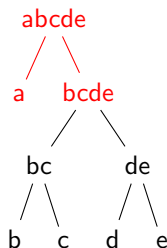
Lista:

**a**: 8/18    **bcde**: 10/18

Menores frequências: **a** e **bcde**

Nova lista:

**abcde**: 18/18

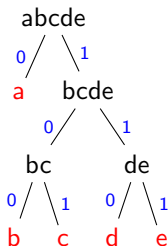


# Huffman: Ilustração

## Codificação de Huffman

Sequência: **abbdacccaabadaeeaa**

Geração da codificação: atribuição de 0 e 1 às arestas



Codificação: **a: 0    b: 100    c: 101    d: 110    e: 111**

Cod. da sequência: **38 bits!**

0 100 100 110 0 101 101 101 0 0 100 0 110 0 111 111 0 0  
a b b d a c c c a a b a d a e e a a

# Otimidade do código de Huffman

## Árvore de custo mínimo

Seja  $T$  uma árvore a partir da qual se produz um **código ótimo** para uma mensagem de  $n$  símbolos.

$$w(T) = \sum_{i=1}^n f_i c_i$$

Sejam  $s_i$  e  $s_j$  dois símbolos de **frequência mínima** com códigos de tamanho  $c_i$  e  $c_j$ , respectivamente. Se  $c_i$  e  $c_j$  não são os maiores códigos, então selecione  $c_l$  e  $c_k$  de tamanho máximo e troque as folhas  $s_k$  e  $s_l$  de  $T$  por  $s_i$  e  $s_j$ , formando uma nova árvore  $T'$ . Como

$$w(T') \leq w(T)$$

Então,  $T$  **pode não ser a melhor árvore!**



# Otimidade do código de Huffman

## Árvore de Huffman é de custo mínimo

Informalmente...

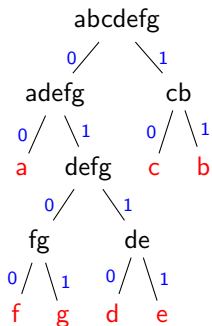
- ▶ A cada passo do algoritmo, os dois símbolos de menor frequência,  $s_i$  e  $s_j$ , são selecionados. A árvore para estes dois símbolos é de custo mínimo, pois estes são folhas cuja distância para a raiz não pode ser maior que aquelas geradas para os demais símbolos.
- ▶ Os dois símbolos  $s_i$  e  $s_j$  são removidos do alfabeto e um novo símbolo é nele inserido, com frequência  $f_i + f_j$ .
- ▶ Repetindo os dois argumentos anteriores, para o novo conjunto de símbolos, pode-se concluir que a árvore gerada é de custo mínimo.

# Exercício

Apresente uma decodificação de Huffman para a sequência:

**dabeeababbebeaeadcccaaddcbbaccaaccaabccffbfdbdg**

**Resposta (uma das alternativas):**



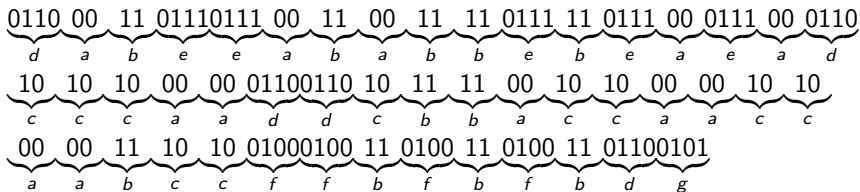
Codificação:

**a: 00    b: 11    c: 10    d: 0110**  
**e: 0111    f: 0100    g: 0101**

# Exercício (Cont. da Resposta)

Codificação da sequência:

**dabeeababbebeaeaddcccaaddcbbaccaaccaabccffbfbfbdg**

0110	00	11	0111	0111	00	11	00	11	11	0111	11	0111	00	0111	00	0110
																
d	a	b	e	e	a	b	a	b	b	e	b	e	a	e	a	d
10	10	10	00	00	0110	0110	10	11	11	00	10	10	00	00	10	10
c	c	c	a	a	d	d	c	b	b	a	c	c	a	a	c	c
00	00	11	10	10	0100	0100	11	0100	11	0100	11	0110	0101			
a	a	b	c	c	f	f	b	f	b	f	b	d	g			

**126 bits**

# Lempel-Ziv-Welch

# Lempel-Ziv-Welch (LZW)

- ▶ Método **baseado em dicionário**: mantém registro sobre sequências de símbolos encontrados ao longo da entrada
- ▶ É **adaptativo**, pois define sequências de símbolos a serem codificadas à medida que se lê a entrada
- ▶ Não requer o armazenamento ou transmissão de tabelas de codificação - requer apenas o conhecimento da codificação de símbolos básicos, previamente conhecidos

- ▶ Inicia-se com uma codificação para o alfabeto de símbolos básicos da entrada
- ▶ Adicionam-se ao dicionário sequências de símbolos (**palavras**) que vão sendo conhecidas

# Codificação - Ilustração

## Exemplo de entrada

**aabaaacaade**

**a a b a a a c a a d e**

↑

**p:**

**a**

Está no dicionário? Sim. Avança

**Saída:**

## Dicionário

**a: 0   d: 3**

**b: 1   e: 4**

**c: 2**

# Codificação - Ilustração

## Exemplo de entrada

**aabaaacaade**

a a b a a a c a a d e  
↑  
**p:** aa Está no dicionário? Não!

**Saída:**

## Dicionário

**a:** 0    **d:** 3  
**b:** 1    **e:** 4  
**c:** 2



# Codificação - Ilustração

## Exemplo de entrada

aabaaacaade

a a b a a a c a a d e  
↑  
**p:** aa Está no dicionário? Não!

Saída:

Insere-se **p** na próxima posição do dicionário; insere-se na saída a palavra de código relativo a **p** sem o último símbolo (**a**); e **p** passa a conter apenas o último símbolo.

## Dicionário

a: 0    d: 3

b: 1    e: 4

c: 2

# Codificação - Ilustração

## Exemplo de entrada

aabaaacaade

a a b a a a c a a d e



**p:** a

Saída: 0

## Dicionário

**a:** 0    **d:** 3

**b:** 1    **e:** 4

**c:** 2    **aa:** 5

# Codificação - Ilustração

## Exemplo de entrada

aabaaacaade

a a b a a a c a a d e  
          ↑  
**p:** ab Está no dicionário? Não!

Saída: 0

Insere-se **p** na próxima posição do dicionário; insere-se na saída a palavra de código relativo a **p** sem o último símbolo (**a**); e **p** passa a conter apenas o último símbolo.

## Dicionário

a: 0    d: 3

b: 1    e: 4

c: 2    aa: 5

# Codificação - Ilustração

## Exemplo de entrada

**aabaaacaade**

**a a b a a a c a a d e**



**p: b**

**Saída: 0 0**

## Dicionário

**a: 0    d: 3    ab: 6**

**b: 1    e: 4**

**c: 2    aa: 5**

# Codificação - Ilustração

## Exemplo de entrada

aabaaacaade

a a b a a a c a a d e  
          ↑  
**p:** ba Está no dicionário? Não!

Saída: 0 0

Insere-se **p** na próxima posição do dicionário; insere-se na saída a palavra de código relativo a **p** sem o último símbolo (**b**); e **p** passa a conter apenas o último símbolo.

## Dicionário

a: 0    d: 3    ab: 6  
b: 1    e: 4  
c: 2    aa: 5

# Codificação - Ilustração

## Exemplo de entrada

**aabaaacaade**

**a a b a a a c a a d e**



**p: a**

**Saída: 0 0 1**

## Dicionário

<b>a: 0</b>	<b>d: 3</b>	<b>ab: 6</b>
<b>b: 1</b>	<b>e: 4</b>	<b>ba: 7</b>
<b>c: 2</b>	<b>aa: 5</b>	

# Codificação - Ilustração

## Exemplo de entrada

**aabaaacaade**

**a a b a a a c a a d e**



**p:** **aa** Está no dicionário? Sim! Avança

**Saída:** 0 0 1

## Dicionário

**a:** 0    **d:** 3    **ab:** 6  
**b:** 1    **e:** 4    **ba:** 7  
**c:** 2    **aa:** 5

# Codificação - Ilustração

## Exemplo de entrada

aabaaacaade

a a b a a a c a a d e



**p:** **aaa** Está no dicionário? Não!

**Saída:** 0 0 1

Insere-se **p** na próxima posição do dicionário; insere-se na saída a palavra de código relativo a **p** sem o último símbolo (**aa**); e **p** passa a conter apenas o último símbolo.

## Dicionário

<b>a:</b> 0	<b>d:</b> 3	<b>ab:</b> 6
<b>b:</b> 1	<b>e:</b> 4	<b>ba:</b> 7
<b>c:</b> 2	<b>aa:</b> 5	



# Codificação - Ilustração

## Exemplo de entrada

aabaaacaade

a a b a a a c a a d e



**p:** a

Saída: 0 0 1 5

## Dicionário

a: 0	d: 3	ab: 6
b: 1	e: 4	ba: 7
c: 2	aa: 5	aaa: 8

# Codificação - Ilustração

## Exemplo de entrada

aabaaacaade

a a b a a a c a a d e



**p:** ac Está no dicionário? Não!

**Saída:** 0 0 1 5

Insere-se **p** na próxima posição do dicionário; insere-se na saída a palavra de código relativo a **p** sem o último símbolo (**a**); e **p** passa a conter apenas o último símbolo.

## Dicionário

**a:** 0    **d:** 3    **ab:** 6

**b:** 1    **e:** 4    **ba:** 7

**c:** 2    **aa:** 5    **aaa:** 8

# Codificação - Ilustração

## Exemplo de entrada

aabaaacaade

a a b a a a c a a d e



p: c

Saída: 0 0 1 5 0

## Dicionário

a: 0	d: 3	ab: 6	ac: 9
b: 1	e: 4	ba: 7	
c: 2	aa: 5	aaa: 8	

# Codificação - Ilustração

## Exemplo de entrada

aabaaacaade

a a b a a a c a a d e

↑  
**p:** ca Está no dicionário? Não!

Saída: 0 0 1 5 0

Insere-se **p** na próxima posição do dicionário; insere-se na saída a palavra de código relativo a **p** sem o último símbolo (**c**); e **p** passa a conter apenas o último símbolo.

## Dicionário

a: 0	d: 3	ab: 6	ac: 9
b: 1	e: 4	ba: 7	
c: 2	aa: 5	aaa: 8	

# Codificação - Ilustração

## Exemplo de entrada

**aabaaacaade**

**a a b a a a c a a d e**



**p: a**

**Saída:** 0 0 1 5 0 2

## Dicionário

<b>a:</b> 0	<b>d:</b> 3	<b>ab:</b> 6	<b>ac:</b> 9
<b>b:</b> 1	<b>e:</b> 4	<b>ba:</b> 7	<b>ca:</b> 10
<b>c:</b> 2	<b>aa:</b> 5	<b>aaa:</b> 8	

# Codificação - Ilustração

## Exemplo de entrada

**aabaaacaade**

a a b a a a c a a d e

**p:** aa      ↑  
Está no dicionário? Sim! Avança

**Saída:** 0 0 1 5 0 2

## Dicionário

<b>a:</b> 0	<b>d:</b> 3	<b>ab:</b> 6	<b>ac:</b> 9
<b>b:</b> 1	<b>e:</b> 4	<b>ba:</b> 7	<b>ca:</b> 10
<b>c:</b> 2	<b>aa:</b> 5	<b>aaa:</b> 8	

# Codificação - Ilustração

## Exemplo de entrada

aabaaacaade

a a b a a a c a a d e

p: aad Está no dicionário? Não!

Saída: 0 0 1 5 0 2

Insere-se **p** na próxima posição do dicionário; insere-se na saída a palavra de código relativo a **p** sem o último símbolo (**aa**); e **p** passa a conter apenas o último símbolo.

## Dicionário

a: 0	d: 3	ab: 6	ac: 9
b: 1	e: 4	ba: 7	ca: 10
c: 2	aa: 5	aaa: 8	

# Codificação - Ilustração

## Exemplo de entrada

aabaaacaade

a a b a a a c a a d e

↑

p: d

Saída: 0 0 1 5 0 2 5

## Dicionário

a: 0	d: 3	ab: 6	ac: 9
b: 1	e: 4	ba: 7	ca: 10
c: 2	aa: 5	aaa: 8	aad: 11



# Codificação - Ilustração

## Exemplo de entrada

aabaaacaade

a a b a a a c a a d e



**p:** de Está no dicionário? Não!

Saída: 0 0 1 5 0 2 5

Insere-se **p** na próxima posição do dicionário; insere-se na saída a palavra de código relativo a **p** sem o último símbolo (**d**); e **p** passa a conter apenas o último símbolo.

## Dicionário

a: 0	d: 3	ab: 6	ac: 9
b: 1	e: 4	ba: 7	ca: 10
c: 2	aa: 5	aaa: 8	aad: 11

# Codificação - Ilustração

## Exemplo de entrada

aabaaacaade

a a b a a a c a a d e  
↑

p: e

Saída: 0 0 1 5 0 2 5 3

## Dicionário

a: 0	d: 3	ab: 6	ac: 9	de: 12
b: 1	e: 4	ba: 7	ca: 10	
c: 2	aa: 5	aaa: 8	aad: 11	

# Codificação - Ilustração

## Exemplo de entrada

aabaaacaade

a a b a a a c a a d e  
↑

**p:** e Está no dicionário? Sim!

Saída: 0 0 1 5 0 2 5 3 4

Como terminou a entrada, insere o código de **p** na saída.

## Dicionário

a: 0	d: 3	ab: 6	ac: 9	de: 12
b: 1	e: 4	ba: 7	ca: 10	
c: 2	aa: 5	aaa: 8	aad: 11	

# Decodificação - Ilustração

Na decodificação, segue-se o **mesmo princípio** da codificação! A partir dos valores da codificação, vai-se gerando a saída e repetindo o processo usado na codificação para montar o dicionário:

	0	0	1	5	0	2	5	3	4
	↑								
Saída:	a								
	↑								
p:	a								

**0** é **a**. Portanto, insere-se o **a** na saída. Há **p** no dicionário? Sim, então segue-se a decodificação

## Dicionário

a: 0    d: 3  
b: 1    e: 4  
c: 2

# Decodificação - Ilustração

Na decodificação, segue-se o **mesmo princípio** da codificação! A partir dos valores da codificação, vai-se gerando a saída e repetindo o processo usado na codificação para montar o dicionário:

	0	0	1	5	0	2	5	3	4
		↑							
Saída:	a	a							
		↑							
p:			aa						

**0** é novamente **a**. Concatena-o em **p**. Há **p** no dicionário? Não! Então, insere-se **p** na próxima posição do dicionário e **p** fica apenas com o último símbolo

## Dicionário

**a:** 0    **d:** 3

**b:** 1    **e:** 4

**c:** 2

# Decodificação - Ilustração

Na decodificação, segue-se o **mesmo princípio** da codificação! A partir dos valores da codificação, vai-se gerando a saída e repetindo o processo usado na codificação para montar o dicionário:

	0	0	1	5	0	2	5	3	4
		↑							
Saída:	a	a							
		↑							
p:	a								

## Dicionário

a: 0    d: 3

b: 1    e: 4

c: 2    aa: 5

# Decodificação - Ilustração

Na decodificação, segue-se o **mesmo princípio** da codificação! A partir dos valores da codificação, vai-se gerando a saída e repetindo o processo usado na codificação para montar o dicionário:

	0	0	1	5	0	2	5	3	4
			↑						
Saída:	a	a	b						
			↑						
<b>p:</b>	ab								

1 é **b**. Coloca-o na saída. Segue-se com **p**. Há **p** no dicionário? Não! Então, insere-se **p** na próxima posição do dicionário e **p** fica apenas com o último símbolo

## Dicionário

**a:** 0    **d:** 3

**b:** 1    **e:** 4

**c:** 2    **aa:** 5

# Decodificação - Ilustração

Na decodificação, segue-se o **mesmo princípio** da codificação! A partir dos valores da codificação, vai-se gerando a saída e repetindo o processo usado na codificação para montar o dicionário:

	0	0	1	5	0	2	5	3	4
			↑						
Saída:	a	a	b						
			↑						
p:	b								

## Dicionário

a: 0	d: 3	ab: 6
b: 1	e: 4	
c: 2	aa: 5	





# Decodificação - Ilustração

Na decodificação, segue-se o **mesmo princípio** da codificação! A partir dos valores da codificação, vai-se gerando a saída e repetindo o processo usado na codificação para montar o dicionário:

	0	0	1	5	0	2	5	3	4
				↑					
Saída:	a	a	b	a	a				
				↑					
p:	a								

## Dicionário

a: 0	d: 3	ab: 6
b: 1	e: 4	ba: 7
c: 2	aa: 5	

# Decodificação - Ilustração

Na decodificação, segue-se o **mesmo princípio** da codificação! A partir dos valores da codificação, vai-se gerando a saída e repetindo o processo usado na codificação para montar o dicionário:

	0	0	1	5	0	2	5	3	4
				↑					
Saída:	a	a	b	a	a				
					↑				
<b>p:</b>	aa								

Avança-se com **p**. Há **p** no dicionário? Sim!

## Dicionário

<b>a:</b> 0	<b>d:</b> 3	<b>ab:</b> 6
<b>b:</b> 1	<b>e:</b> 4	<b>ba:</b> 7
<b>c:</b> 2	<b>aa:</b> 5	

# Decodificação - Ilustração

Na decodificação, segue-se o **mesmo princípio** da codificação! A partir dos valores da codificação, vai-se gerando a saída e repetindo o processo usado na codificação para montar o dicionário:

	0	0	1	5	0	2	5	3	4
					↑				
Saída:	a	a	b	a	a	a			
						↑			
<b>p:</b>	aaa								

**0** é **a**. Coloca-o na saída. Segue-e com **p**. Há **p** no dicionário? Não! Então, insere-se **p** na próxima posição do dicionário e **p** fica apenas com o último símbolo

## Dicionário

<b>a:</b> 0	<b>d:</b> 3	<b>ab:</b> 6
<b>b:</b> 1	<b>e:</b> 4	<b>ba:</b> 7
<b>c:</b> 2	<b>aa:</b> 5	

# Decodificação - Ilustração

Na decodificação, segue-se o **mesmo princípio** da codificação! A partir dos valores da codificação, vai-se gerando a saída e repetindo o processo usado na codificação para montar o dicionário:

	0	0	1	5	0	2	5	3	4
					↑				
Saída:	a	a	b	a	a	a			
						↑			
p:	a								

## Dicionário

a: 0	d: 3	ab: 6
b: 1	e: 4	ba: 7
c: 2	aa: 5	aaa: 8

# Decodificação - Ilustração

Na decodificação, segue-se o **mesmo princípio** da codificação! A partir dos valores da codificação, vai-se gerando a saída e repetindo o processo usado na codificação para montar o dicionário:

	0	0	1	5	0	2	5	3	4
						↑			
Saída:	a	a	b	a	a	a	c		
							↑		
<b>p:</b>							ac		

2 é **c**. Coloca-o na saída. Segue-se com **p**. Há **p** no dicionário? Não! Então, insere-se **p** na próxima posição do dicionário e **p** fica apenas com o último símbolo

## Dicionário

<b>a:</b> 0	<b>d:</b> 3	<b>ab:</b> 6
<b>b:</b> 1	<b>e:</b> 4	<b>ba:</b> 7
<b>c:</b> 2	<b>aa:</b> 5	<b>aaa:</b> 8

# Decodificação - Ilustração

Na decodificação, segue-se o **mesmo princípio** da codificação! A partir dos valores da codificação, vai-se gerando a saída e repetindo o processo usado na codificação para montar o dicionário:

	0	0	1	5	0	2	5	3	4
						↑			
Saída:	a	a	b	a	a	a	c		
							↑		
p:	c								

## Dicionário

a: 0	d: 3	ab: 6	ac: 9
b: 1	e: 4	ba: 7	
c: 2	aa: 5	aaa: 8	

# Decodificação - Ilustração

Na decodificação, segue-se o **mesmo princípio** da codificação! A partir dos valores da codificação, vai-se gerando a saída e repetindo o processo usado na codificação para montar o dicionário:

	0	0	1	5	0	2	5	3	4
							↑		
Saída:	a	a	b	a	a	a	c	a	a
								↑	
<b>p:</b>	ca								

5 é **aa**. Coloca-os na saída. Segue-se com **p**. Há **p** no dicionário? Não! Então, insere-se **p** na próxima posição do dicionário e **p** fica apenas com o último símbolo

## Dicionário

a: 0	d: 3	ab: 6	ac: 9
b: 1	e: 4	ba: 7	
c: 2	aa: 5	aaa: 8	



# Decodificação - Ilustração

Na decodificação, segue-se o **mesmo princípio** da codificação! A partir dos valores da codificação, vai-se gerando a saída e repetindo o processo usado na codificação para montar o dicionário:

	0	0	1	5	0	2	5	3	4
							↑		
Saída:	a	a	b	a	a	a	c	a	a
								↑	
p:	a								

## Dicionário

a: 0	d: 3	ab: 6	ac: 9
b: 1	e: 4	ba: 7	ca: 10
c: 2	aa: 5	aaa: 8	

# Decodificação - Ilustração

Na decodificação, segue-se o **mesmo princípio** da codificação! A partir dos valores da codificação, vai-se gerando a saída e repetindo o processo usado na codificação para montar o dicionário:

	0	0	1	5	0	2	5	3	4
							↑		
Saída:	a	a	b	a	a	a	c	a	a
								↑	
<b>p:</b>	aa								

Segue-se com **p**. Há **p** no dicionário? Sim!

## Dicionário

<b>a:</b> 0	<b>d:</b> 3	<b>ab:</b> 6	<b>ac:</b> 9
<b>b:</b> 1	<b>e:</b> 4	<b>ba:</b> 7	<b>ca:</b> 10
<b>c:</b> 2	<b>aa:</b> 5	<b>aaa:</b> 8	

# Decodificação - Ilustração

Na decodificação, segue-se o **mesmo princípio** da codificação! A partir dos valores da codificação, vai-se gerando a saída e repetindo o processo usado na codificação para montar o dicionário:

	0	0	1	5	0	2	5	3	4	
								↑		
Saída:	a	a	b	a	a	a	c	a	a	d
									↑	
<b>p:</b>	aad									

3 é **d**. Coloca-o na saída. Segue-se com **p**. Há **p** no dicionário? Não! Então, insere-se **p** na próxima posição do dicionário e **p** fica apenas com o último símbolo

## Dicionário

<b>a:</b> 0	<b>d:</b> 3	<b>ab:</b> 6	<b>ac:</b> 9
<b>b:</b> 1	<b>e:</b> 4	<b>ba:</b> 7	<b>ca:</b> 10
<b>c:</b> 2	<b>aa:</b> 5	<b>aaa:</b> 8	

# Decodificação - Ilustração

Na decodificação, segue-se o **mesmo princípio** da codificação! A partir dos valores da codificação, vai-se gerando a saída e repetindo o processo usado na codificação para montar o dicionário:

	0	0	1	5	0	2	5	3	4	
								↑		
Saída:	a	a	b	a	a	a	c	a	a	d
									↑	
p:	d									

## Dicionário

a: 0	d: 3	ab: 6	ac: 9
b: 1	e: 4	ba: 7	ca: 10
c: 2	aa: 5	aaa: 8	aad: 11

# Decodificação - Ilustração

Na decodificação, segue-se o **mesmo princípio** da codificação! A partir dos valores da codificação, vai-se gerando a saída e repetindo o processo usado na codificação para montar o dicionário:

	0	0	1	5	0	2	5	3	4	
								↑		
Saída:	a	a	b	a	a	a	c	a	a	d e
									↑	
p:										d

**4** é **e**. Coloca-o na saída. Termina-se a decodificação

## Dicionário

a: 0	d: 3	ab: 6	ac: 9
b: 1	e: 4	ba: 7	ca: 10
c: 2	aa: 5	aaa: 8	aad: 11

# Decodificação - Caso Especial

Suponha que se queira decodificar a sequência abaixo, considerando um alfabeto apenas com as letras **a** (0) e **b** (1)

0 1 2 4 1  
↑  
Saída: a  
↑  
**p**: a

0 é **a**. Coloca-o na saída. Há **p** no dicionário? Sim!

## Dicionário

**a**: 0

**b**: 1

# Decodificação - Caso Especial

Suponha que se queira decodificar a sequência abaixo, considerando um alfabeto apenas com as letras **a** (0) e **b** (1)

	0	1	2	4	1
		↑			
Saída:	a	b			
		↑			
<b>p:</b>	ab				

**1** é **b**. Coloca-o na saída. Há **p** no dicionário? Não! Então, insere-se **p** na próxima posição do dicionário e **p** fica apenas com o último símbolo

## Dicionário

**a:** 0

**b:** 1

# Decodificação - Caso Especial

Suponha que se queira decodificar a sequência abaixo, considerando um alfabeto apenas com as letras **a** (0) e **b** (1)

	0	1	2	4	1
		↑			
Saída:	a	b			
		↑			
p:	b				

## Dicionário

**a:** 0

**b:** 1

**ab:** 2



# Decodificação - Caso Especial

Suponha que se queira decodificar a sequência abaixo, considerando um alfabeto apenas com as letras **a** (0) e **b** (1)

	0	1	2	4	1
			↑		
Saída:	a	b	a	b	
			↑		
<b>p:</b>	ba				

2 é **ab**. Coloca-os na saída. Há **p** no dicionário? Não! Então, insere-se **p** na próxima posição do dicionário e **p** fica apenas com o último símbolo

## Dicionário

**a:** 0

**b:** 1

**ab:** 2

# Decodificação - Caso Especial

Suponha que se queira decodificar a sequência abaixo, considerando um alfabeto apenas com as letras **a** (0) e **b** (1)

	0	1	2	4	1
			↑		
Saída:	a	b	a	b	
			↑		
p:	a				

## Dicionário

**a:** 0    **ba:** 3

**b:** 1

**ab:** 2

# Decodificação - Caso Especial

Suponha que se queira decodificar a sequência abaixo, considerando um alfabeto apenas com as letras **a** (0) e **b** (1)

	0	1	2	4	1
			↑		
Saída:	a	b	a	b	
				↑	
<b>p:</b>	ab				

Segue-se com **p**. Há **p** no dicionário? Sim! Vamos prosseguir!

## Dicionário

**a:** 0    **ba:** 3

**b:** 1

**ab:** 2

# Decodificação - Caso Especial

Suponha que se queira decodificar a sequência abaixo, considerando um alfabeto apenas com as letras **a** (0) e **b** (1)

	0	1	2	4	1
			↑		
Saída:	a	b	a	b	
				↑	
p:	ab				

**Ops! Não há 4 no dicionário! O que fazer?**

## Dicionário

**a:** 0    **ba:** 3

**b:** 1

**ab:** 2

# Decodificação - Caso Especial

Suponha que se queira decodificar a sequência abaixo, considerando um alfabeto apenas com as letras **a** (0) e **b** (1)

	0	1	2	4	1
			↑		
Saída:	a	b	a	b	
				↑	
p:	ab				

**Ops! Não há 4 no dicionário! O que fazer?**

É possível saber qual é a entrada 4. Como?

## Dicionário

**a:** 0    **ba:** 3

**b:** 1

**ab:** 2

# Decodificação - Caso Especial

Suponha que se queira decodificar a sequência abaixo, considerando um alfabeto apenas com as letras **a** (0) e **b** (1)

	0	1	2	4	1
			↑		
Saída:	a	b	a	b	
				↑	
p:	ab				

A entrada 4 somente pode ser **aba**!

## Dicionário

**a:** 0    **ba:** 3

**b:** 1

**ab:** 2

# Decodificação - Caso Especial

Suponha que se queira decodificar a sequência abaixo, considerando um alfabeto apenas com as letras **a** (0) e **b** (1)

	0	1	2	4	1
			↑		
Saída:	a	b	a	b	
				↑	
p:	ab				

A entrada 4 somente pode ser **aba**!

Veja que, na codificação há o **2**, relativo a **ab**. Neste ponto, na codificação, será criada a entrada **4**.

## Dicionário

**a:** 0    **ba:** 3

**b:** 1

**ab:** 2

# Decodificação - Caso Especial

Suponha que se queira decodificar a sequência abaixo, considerando um alfabeto apenas com as letras **a** (0) e **b** (1)

	0	1	2	4	1
Saída:	a	b	a	b	X

↑

↑

**p:** abX

Como ela é criada? Ela será da forma **abX**, sendo **X** o próximo caractere.

## Dicionário

<b>a:</b> 0	<b>ba:</b> 3
<b>b:</b> 1	<b>abX:</b> 4
<b>ab:</b> 2	



# Decodificação - Caso Especial

Suponha que se queira decodificar a sequência abaixo, considerando um alfabeto apenas com as letras **a** (0) e **b** (1)

	0	1	2	4	1
			↑		
Saída:	a	b	a b	X	
	⏟ 0	⏟ 1	⏟ 2	⏟ 4	

**p:** abX

Mas qual caractere é **X**? Considerando que o próximo número da codificação é **4**, a sequência que começa após o **2** é **abX**. Portanto **X** é **a**!

## Dicionário

a: 0      ba: 3  
b: 1      abX: 4  
ab: 2

# Decodificação - Caso Especial

Suponha que se queira decodificar a sequência abaixo, considerando um alfabeto apenas com as letras **a** (0) e **b** (1)

	0	1	2		4		1	
			↑					
Saída:	a	b	a	b	X			
	a	b	a	b	a	b	X	
	⏟ 0		⏟ 1		⏟ 2		⏟ 4	
p:	abX							

Mas qual caractere é **X**? Considerando que o próximo número da codificação é **4**, a sequência que começa após o **2** é **abX**. Portanto **X** é **a**!

## Dicionário

a: 0    ba: 3  
b: 1    abX: 4  
ab: 2

# Decodificação - Caso Especial

Suponha que se queira decodificar a sequência abaixo, considerando um alfabeto apenas com as letras **a** (0) e **b** (1)

Saída:       $\underbrace{\quad}_0$      $\underbrace{\quad}_1$      $\underbrace{\quad}_2$      $\underbrace{\quad}_4$      $\underbrace{\quad}_1$   
          **a**        **b**        **a** **b**    **a** **b** **a**    **b**

Com isso, seguindo-se o algoritmo, chega-se à decodificação final

## Dicionário

**a:** 0      **ba:** 3

**b:** 1      **aba:** 4

**ab:** 2