

BCM0504

# Natureza da Informação

## Compressão

Prof. Alexandre Donizeti Alves



Universidade Federal do ABC

Bacharelado em Ciência e Tecnologia

Bacharelado em Ciências e Humanidades

Terceiro Quadrimestre - 2018

# Códigos

- Representação de objetos complexos
  - **Exemplo:** por cadeias (padrões) de bits
  - Focaremos o uso de cadeias de bits
- **Código:** mapeamento entre objetos a serem representados (símbolos) e as cadeias de bits usadas para representá-los

Queremos códigos **pequenos** e **pouco suscetíveis a erros**

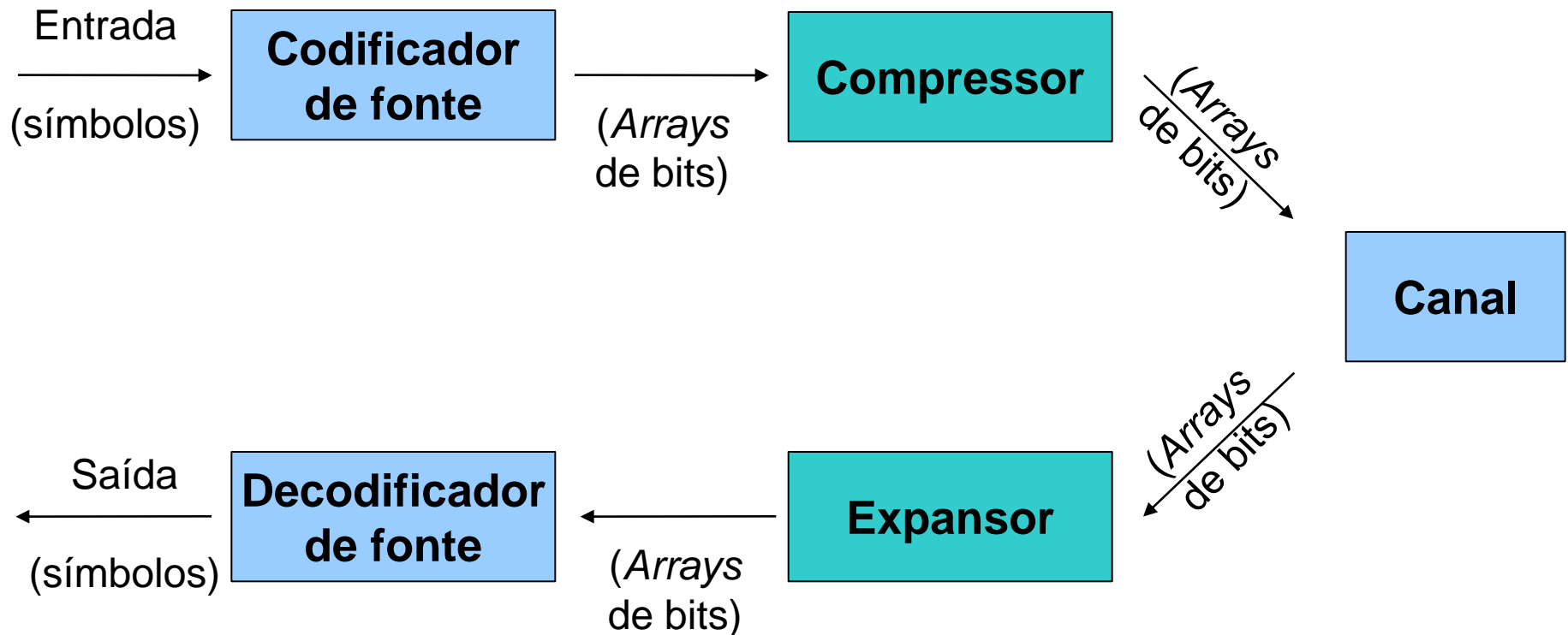
**Veremos técnicas de compressão para geração de representações eficientes**

# Compressão

- **Objetivo:** converter cadeia de bits representando símbolos sucessivos em uma cadeia menor para economia em:
  - ❑ Transmissão
  - ❑ Armazenamento
  - ❑ Processamento
- Diminui mensagens, mas **aumenta processamento** necessário

# Modelo de sistema de comunicação

## ■ Modelo estendido



# Tipos de compressão

- **Sem perda** ou **reversível**
  - Símbolo original/codificado pode ser reconstruído exatamente a partir da versão comprimida
  - Usualmente explora redundâncias
    - **Exemplo:** tem padrões de bits não usados ou tem alguns símbolos mais usados que outros

# Tipos de compressão

- **Com perda** ou **irreversível**

- Símbolo original/codificado não pode ser reconstruído exatamente a partir da versão comprimida

- **Expansor** produz aproximação

# Tipos de compressão

- **Exemplo:** mensagem 25.8888888888
  - Pode ser comprimida **sem perda** como: 25.[9]8
    - “vinte e cinco ponto 9 oitos”
    - Cadeia original pode ser perfeitamente recriada
  - Ou comprimida **com perda** como: 26
    - O dado exato é perdido, porém há o benefício de uma codificação menor

# Técnicas de compressão sem perda

- Permitem reconstrução exata da mensagem:
    - ❑ Codificação de tamanho variável
    - ❑ Codificação *run-length*
    - ❑ Dicionário estático
    - ❑ Dicionário semi-adaptativo
    - ❑ Dicionário dinâmico
-



# Codificação de tamanho variável

- Atribuição de códigos menores a símbolos mais frequentes
- E códigos maiores para símbolos menos frequentes
- Pode ser feita no codificador da fonte ou no compressor
  - **Procedimento geral será discutido em aula posterior**

# Codificação *run-length*

- Suponha mensagem com sequências longas de repetições de símbolos
  - Pode codificar mensagem como lista do símbolo e número de vezes que ele ocorre
    - **Exemplo:** mensagem “a B B B B B B a a a B B a a a a” poderia ser codificada como “a[1] B[5] a[3] B[2] a[4]”
  - Funciona bem em poucas circunstâncias

# Codificação *run-length*

## ■ Exemplos:

### □ Bandeira da Alemanha

- X pixels pretos, X pixels vermelhos, X pixels amarelos



### □ Fax

- Documento é escaneado em preto e branco
- Longos grupos de pixels brancos ou pretos são transmitidos como o número desses pixels



### □ Otimização de uso de discos em computadores



# Codificação *run-length*

- Não funciona bem para mensagens sem repetições de sequências de um mesmo símbolo
  - ❑ **Exemplo:** fotografias, em que pequenas mudanças em sombreamento de um pixel a outro requer o uso de vários símbolos



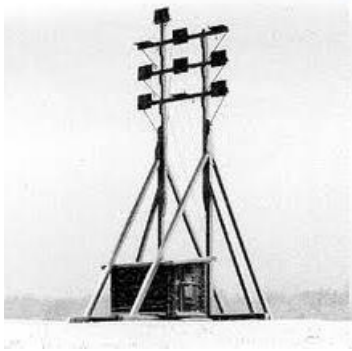
# Dicionário estático

- Se um código tem padrões não usados, eles podem ser atribuídos a sequências que ocorrem frequentemente
  - Como abreviações
  - E essas sequências passam a requerer menos bits do que a transmissão símbolo a símbolo
    - Exemplo: DEL em ASCII pode ser desnecessário
      - $\Rightarrow$  atribuir código 127 à palavra “the”
  - Lista de códigos e seus significados é chamada livro de códigos ou **dicionário**

# Dicionário estático

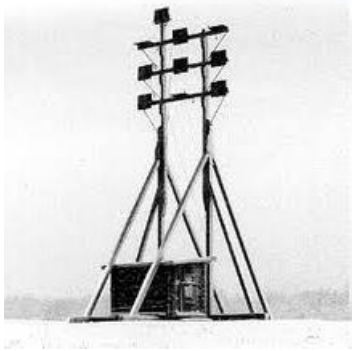
## ■ Exemplo:

- Telégrafo mecânico da marinha britânica (1796)
  - Comunicação entre sede e portos
  - Séries de guaritas, cada uma com visão para a próxima
    - Cada uma com seis grandes paletas



# Dicionário estático

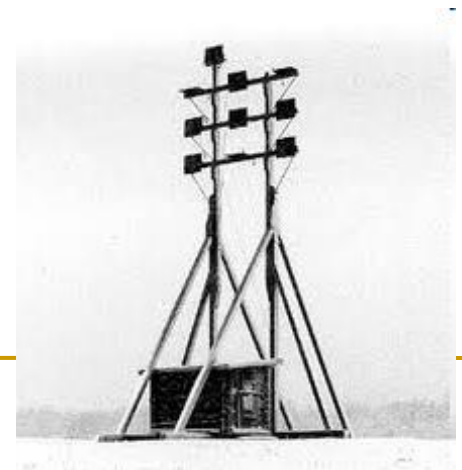
- ❑ Cada uma com seis grandes paletas
  - Podiam ser rotacionadas para posição horizontal (aberto) e vertical (fechado)
  - **Operação:** todas abertas até ter que mandar uma mensagem, quando todas eram fechadas, sinalizando início de mensagem
  - Os operadores deveriam checar se havia mensagem a cada 5 minutos
  - **Mensagem:** sequência de padrões das paletas, terminando com todas abertas



# Dicionário estático

## ■ Exemplo:

- ❑ Telégrafo mecânico da marinha britânica (1796)
  - Seis paletas, dois símbolos cada  $\Rightarrow 2^6 = 64$  padrões
    - ❑ Dois deles: de controle
      - Todos abertos = começo e final
      - Todos fechados = inativo
    - ❑ Outros 62:
      - 24 letras do alfabeto Inglês da época (sem J e U)
      - 10 dígitos
      - Marcador de fim de palavra
      - Marcador de fim de página
  - ❑ **Sobram 20**

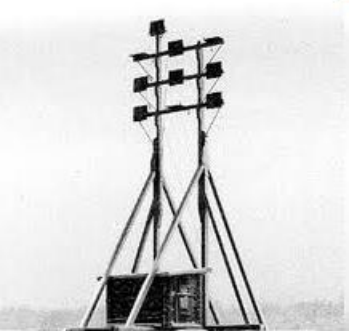




# Dicionário estático

## ■ Exemplo:

- Telégrafo mecânico da marinha britânica (1796)
  - 20 padrões não usados foram sendo atribuídos a palavras ou frases frequentes
    - **Exemplo:** “*the*”, localidades, “*east*”, “*Sentence of court-martial to be put in execution*” etc.
    - Assim, mensagens grandes podiam ser reduzidas consideravelmente com escolhas boas de abreviações
    - **Mas havia riscos:** os efeitos de erros podem ser maiores
      - Transmissão de texto completo: erros pontuais podem ser detectados pelo humano
      - Abreviações: um único bit incorreto pode resultar em um significado indesejado



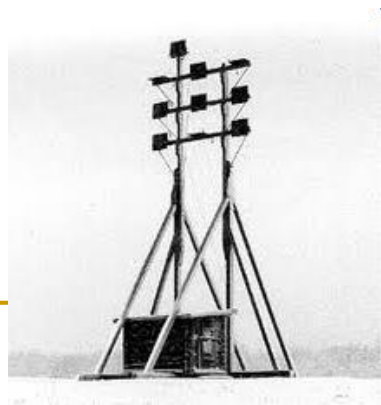
# Dicionário estático

## ■ Exemplo:

- Telégrafo mecânico da marinha britânica (1796)

## ■ Funcionou bem na época

- Uma mensagem foi transmitida a uma distância de aproximadamente 800 km em 3 minutos



# Dicionário estático

- Uso de abreviações para comprimir mensagens implica **necessidade de dicionário**
    - **E de transmiti-lo antes da primeira mensagem**
      - Deve ser cuidadosamente construído e **não pode ser alterado com frequência**
    - Bom para conjuntos de mensagens similares
    - Não adequado para mensagens diversas
-

# Dicionário semi-adaptativo

- Se um novo dicionário pudesse ser definido para cada mensagem, a compressão seria maior
  - Sequências de símbolos da mensagem em particular poderiam ser entradas do dicionário
- **Problemas:**
  - Novo dicionário teria que ser transmitido junto à mensagem
  - Mensagem deve ser analisada antes para descobrir as melhores entradas para o dicionário
    - Deve ser conhecida antes e armazenada como um todo antes

**Uso limitado**

# Dicionário dinâmico

- Calculado à medida que a mensagem é processada
- Não precisa acompanhar a mensagem
- Pode ser usado antes do fim da mensagem ser processado

# Dicionário dinâmico

## ■ Técnica de compressão LZW

□ Abraham Lempel, Jacob Ziv, Terry Welch



- Lempel e Ziv: várias técnicas de compressão, LZ77 e LZ78
- Modificações de Welch em 1984

# Dicionário dinâmico

- Welch queria reduzir número de bits enviados a cabeça de gravação em discos
  - Aumentar capacidade dos discos
  - Melhorar velocidade de transmissão de dados
- Aplicado a arquivos de:
  - Textos: reduz tipicamente a metade do tamanho
  - Imagens, com grandes áreas com mesma cor: pode reduzir a menos da metade
    - GIF usa compressão LWZ
    - Em fotos com mudanças graduais de cor, compressão é muito mais modesta

**Problema: patente até 2004**

# Compressão LZW

- **Reversível**: não perde informação, decodificador reconstrói mensagem original exatamente
- **Dicionário inicial**: 8 bits (0 a 255)
  - ASCII nos primeiros caracteres
- **Entrada 256**: começo da transmissão (limpar dicionário)
- **Entrada 257**: fim de transmissão ou parada
- **Mensagem codificada** = sequência de números
  - Códigos representando as entradas no dicionário



# Compressão LZW

- Dicionário inicialmente contém caracteres isolados, mas **novas entradas com dois ou mais caracteres** vão sendo definidas a medida que a mensagem é analisada

# Compressão LZW

## ■ Codificação:

- **nova\_entrada** = novas entradas do  
dicionário =  $\phi$
- Manda código de início (256)
- Adiciona a **nova\_entrada** os caracteres, um  
a um, da mensagem sendo comprimida

# Compressão LZW

## ■ Codificação:

- Assim que **nova\_entrada** se torna diferente de qualquer outra cadeia já no dicionário, coloca-a no dicionário e manda código da cadeia sem o último caractere
- Quando mensagem termina, manda código de **nova\_entrada** e código de parada (257)

# Compressão LZW

## ■ Decodificação:

- ❑ Se recebe código de início (256), limpa dicionário novo e seta **nova\_entrada'** =  $\phi$
- ❑ Para o código seguinte recebido, produz como saída o caractere representado pelo código e coloca-o em **nova\_entrada'**

# Compressão LZW

## ■ Decodificação:

- Para códigos subsequentes recebidos

- ***Se conhece o código:***

- Adiciona o primeiro caractere da cadeia representada pelo código **nova\_entrada'**, insere resultado no dicionário e produz como saída a cadeia do código recebido

# Compressão LZW

## ■ Decodificação:

- ❑ Coloca em **nova\_entrada'** a cadeia anterior, para começar nova entrada do dicionário

# Compressão LZW

## ■ Decodificação:

### ■ *Senão:*

- ❑ Repete em **nova\_entrada'** primeiro caractere do código recebido anteriormente, insere resultado no dicionário e produz como saída a cadeia do código recebido
  - ❑ Quando o código de parada (257) é recebido, nada precisa ser feito
-

# Exemplo 1: mensagem “itty bitty bit bin”

Codificação

Transmissão

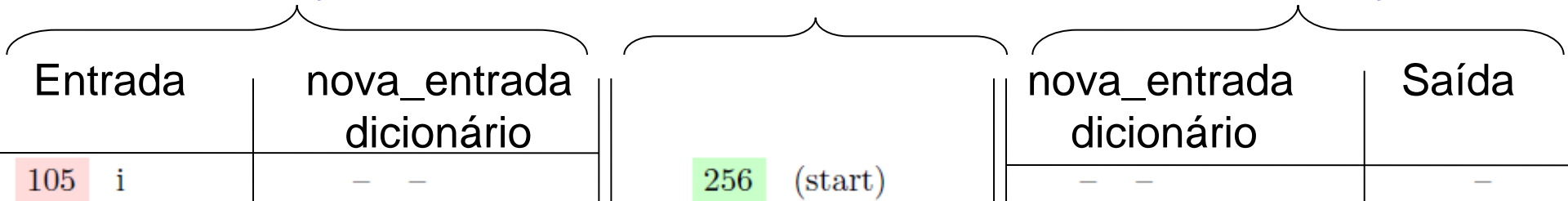
Decodificação

Entrada		nova_entrada dicionário				nova_entrada dicionário		Saída
105	i	—	—	256	(start)	—	—	—
116	t	258	i t	105	i	—	—	i
116	t	259	t t	116	t	258	i t	t
121	y	260	t y	116	t	259	t t	t
32	space	261	y space	121	y	260	t y	y
98	b	262	space b	32	space	261	y space	space
105	i	263	b i	98	b	262	space b	b
116	t	—	—	—	—	—	—	—
116	t	264	i t t	258	i t	263	b i	i t
121	y	—	—	—	—	—	—	—
32	space	—	t y space	260	t y	264	i t t	t y
98	b	Entrada: caracteres com 8 bits (código ASCII)		—	—	Transmissão: caracteres com 9 bits		—
105	i			262	space b			space b
116	t	—	—	—	—	258	i t	i t
32	space	—	—	258	i t	—	—	—
98	b	—	—	—	—	266	space b i	—
105	i	—	—	—	—	—	—	—
110	n	268	space b i n	266	space b i	267	i t space	space b i
—	—	—	—	110	n	268	space b i n	n
—	—	—	—	257	(stop)	—	—	—



## ■ Exemplo 1: mensagem “*itty bitty bit bin*”

Codificação → Transmissão → Decodificação



1) Nova\_entrada =  $\phi$

4) Nova\_entrada = i

2) Manda  
código de início

3) Recebendo  
código de início,  
limpa dicionário e faz  
nova\_entrada' =  $\phi$

# ■ Exemplo 1: mensagem “*itty bitty bit bin*”

Codificação → Transmissão → Decodificação

Entrada		nova_entrada dicionário				nova_entrada dicionário		Saída
105	i	-	-	256	(start)	-	-	-
116	t	258	i t	105	i	-	-	i

1) Nova\_entrada = i t  
(não tem no dicionário)  
⇒ 2) Insere no dicionário

3) Manda  
código de  
nova\_entrada,  
sem o caractere  
final ⇒ i

5) Produz como saída  
o caractere do código  
recebido ⇒ i  
6) Nova\_entrada' = i

4) nova\_entrada=t

■ Exemplo 1: mensagem “*itty bitty bit bin*”

Codificação

Transmissão

Decodificação

Entrada		nova_entrada dicionário				nova_entrada dicionário		Saída
105	i	—	—	256	(start)	—	—	—
116	t	258	i t	105	i	—	—	i
116	t	259	t t	116	t	258	i t	t

- 1) Nova\_entrada = t  
(tem no dicionário)
- 2) Nova\_entrada = t t  
(não tem)
- ⇒ 3) Insere no dicionário

- 4) Manda  
código de  
nova\_entrada,  
sem o caractere  
final ⇒ t

- 5) nova\_entrada=t

- 6) Nova\_entrada' = i t
- 7) Insere  
nova\_entrada' no  
dicionário
- 8) Produz saída =  
cadeia do código  
recebido ⇒ t

- 9) nova\_entrada'=t

# Exemplo 1: mensagem “*itty bitty bit bin*”

Codificação → Transmissão → Decodificação

Entrada		nova_entrada dicionário		nova_entrada dicionário		Saída
105	i	—	—	256	(start)	—
116	t	258	i t	105	i	i
116	t	259	t t	116	t	t
121	y	260	t y	116	t	t

- 1) Nova\_entrada = t  
(tem no dicionário)
- 2) Nova\_entrada = t y  
(não tem)
- ⇒ 3) Insere no dicionário

5) nova\_entrada=y

- 4) Manda  
código de  
nova\_entrada,  
sem o caractere  
final ⇒ t

- 6) Nova\_entrada' = t t
- 7) Insere  
nova\_entrada' no  
dicionário
- 8) Produz saída =  
cadeia do código  
recebido ⇒ t

9) nova\_entrada'=t

# Exemplo 1: mensagem “*itty\_bitty bit bin*”

Codificação → Transmissão → Decodificação

Entrada		nova_entrada dicionário		nova_entrada dicionário		Saída
105	i	—	—	256	(start)	—
116	t	258	i t	105	i	i
116	t	259	t t	116	t	t
121	y	260	t y	116	t	t
32	space	261	y space	121	y	y

1) Nova\_entrada = y  
(tem no dicionário)  
2) Nova\_entrada =  
y space (não tem)  
⇒ 3) Insere no dicionário

5) nova\_entrada=space

4) Manda  
código de  
nova\_entrada,  
sem o caractere  
final ⇒ y

6) Nova\_entrada' = t y  
7) Insere  
nova\_entrada' no  
dicionário  
8) Produz saída =  
cadeia do código  
recebido ⇒ y

9) nova\_entrada'=y

# Exemplo 1: mensagem “*itty\_bitty bit bin*”

Codificação

Transmissão

Decodificação

Entrada		nova_entrada dicionário		nova_entrada dicionário		Saída
105	i	—	—	—	—	—
116	t	258	i t	—	—	i
116	t	259	t t	258	i t	t
121	y	260	t y	259	t t	t
32	space	261	y space	260	t y	y
98	b	262	space b	261	y space	space

- 1) Nova\_entrada = space (tem no dicionário)
- 2) Nova\_entrada = space b (não tem)
- ⇒ 3) Insere no dicionário

5) nova\_entrada=b

- 4) Manda código de nova\_entrada, sem o caractere final ⇒ space

- 6) Nova\_entrada' = y space
- 7) Insere nova\_entrada' no dicionário
- 8) Produz saída = cadeia do código recebido ⇒ space

9) nova\_entrada'=space

# Exemplo 1: mensagem “*itty\_bitty bit bin*”

Codificação

Transmissão

Decodificação

Entrada		nova_entrada dicionário				nova_entrada dicionário		Saída
105	i	—	—	256	(start)	—	—	—
116	t	258	i t	105	i	—	—	i
116	t	259	t t	116	t	258	i t	t
121	y	260	t y	116	t	259	t t	t
32	space	261	y space	121	y	260	t y	y
98	b	262	space b	32	space	261	y space	space
105	i	263	b i	98	b	262	space b	b

1) Nova\_entrada = b  
(tem no dicionário)  
2) Nova\_entrada =  
b i (não tem)  
⇒ 3) Insere no dicionário

5) nova\_entrada=i

4) Manda  
código de  
nova\_entrada,  
sem o caractere  
final ⇒ b

6) Nova\_entrada' =  
space b  
7) Insere  
nova\_entrada' no  
dicionário  
8) Produz saída =  
cadeia do código  
recebido ⇒ b

9) nova\_entrada'=b

# ■ Exemplo 1: mensagem “*itty\_bitty bit bin*”

Codificação → Transmissão → Decodificação

Entrada		nova_entrada dicionário		Transmissão		nova_entrada dicionário		Saída
105	i	—	—	256	(start)	—	—	—
116	t	258	i t	105	i	—	—	i
116	t	259	t t	116	t	258	i t	t
121	y	260	t y	116	t	259	t t	t
32	space	261	y space	121	y	260	t y	y
98	b	262	space b	32	space	261	y space	space
105	i	263	b i	98	b	262	space b	b
116	t	—	—	—	—	—	—	—

- 1) Nova\_entrada = i  
(tem no dicionário)
- 2) Nova\_entrada =  
i t (tem)



■ Exemplo 1: mensagem “*itty\_bitty bit bin*”

Codificação → Transmissão → Decodificação

Entrada		nova_entrada dicionário		nova_entrada dicionário		Saída
105	i	—	—	256	(start)	—
116	t	258	i t	105	i	i
116	t	259	t t	116	t	t
121	y	260	t y	116	t	t
32	space	261	y space	121	y	y
98	b	262	space b	32	space	space
105	i	263	b i	98	b	b
116	t	—	—	—	—	—
116	t	264	i t t	258	i t	i t

1) Nova\_entrada = i t  
(tem no dicionário)  
2) Nova\_entrada =  
i t t (não tem)  
⇒ 3) Insere no dicionário

5) nova\_entrada=t

4) Manda  
código de  
nova\_entrada,  
sem o caractere  
final ⇒ i t

6) Nova\_entrada' = b i  
7) Insere nova\_entrada'  
no dicionário  
8) Produz saída = cadeia  
do código recebido ⇒ i t

9) nova\_entrada'=i t

■ Exemplo 1: mensagem “*itty\_bitty bit bin*”

Codificação → Transmissão → Decodificação

Entrada		nova_entrada dicionário		nova_entrada dicionário		Saída
105	i	—	—	256	(start)	—
116	t	258	i t	105	i	i
116	t	259	t t	116	t	t
121	y	260	t y	116	t	t
32	space	261	y space	121	y	y
98	b	262	space b	32	space	space
105	i	263	b i	98	b	b
116	t	—	—	—	—	—
116	t	264	i t t	258	i t	i t
121	y	—	—	—	—	—

- 1) Nova\_entrada = t  
(tem no dicionário)
- 2) Nova\_entrada =  
t y (tem)

# Exemplo 1: mensagem “*itty\_bitty\_bit bin*”

Codificação → Transmissão → Decodificação

Entrada		nova_entrada dicionário		Transmissão		nova_entrada dicionário		Saída
105	i	—	—	256	(start)	—	—	—
116	t	258	i t	105	i	—	—	i
116	t	259	t t	116	t	258	i t	t
121	y	260	t y	116	t	259	t t	t
32	space	261	y space	121	y	260	t y	y
98	b	262	space b	32	space	261	y space	space
105	i	263	b i	98	b	262	space b	b
116	t	—	—	—	—	—	—	—
116	t	264	t t t	258	i t	—	—	—
121	y	—	—	—	—	—	—	—
32	space	265	t y space	260	t y	264	i t t	t y

5) nova\_entrada=space

9) nova\_entrada'=t y

1) Nova\_entrada = t y  
(tem no dicionário)  
2) Nova\_entrada =  
t y space (não tem)  
⇒ 3) Insere no dicionário

4) Manda  
código de  
nova\_entrada,  
sem o caractere  
final ⇒ t y

6) Nova\_entrada' = i t t  
7) Insere nova\_entrada'  
no dicionário  
8) Produz saída = cadeia  
do código recebido ⇒ t y

# Exemplo 1: mensagem “*itty\_bitty\_bit bin*”

Codificação → Transmissão → Decodificação

Entrada		nova_entrada dicionário		Transmissão		nova_entrada dicionário		Saída
105	i	—	—	256	(start)	—	—	—
116	t	258	i t	105	i	—	—	i
116	t	259	t t	116	t	258	i t	t
121	y	260	t y	116	t	259	t t	t
32	space	261	y space	121	y	260	t y	y
98	b	262	space b	32	space	261	y space	space
105	i	263	b i	98	b	262	space b	b
116	t	—	—	—	—	—	—	—
116	t	264	i t t	258	i t	263	b i	i t
121	y	—	—	—	—	—	—	—
32	space	265	t y space	260	t y	264	i t t	t y
98	b	—	—	—	—	—	—	—

- 1) Nova\_entrada = space  
(tem no dicionário)
- 2) Nova\_entrada =  
space b (tem)

# Exemplo 1: mensagem “*itty\_bitty\_bit bin*”

Codificação

Transmissão

Decodificação

Entrada		nova_entrada dicionário		nova_entrada dicionário		Saída
105	i	-	-	256	(start)	-
116	t	258	i t	105	i	i
116	t	259	t t	116	t	t
121	y	260	t y	116	t	t
32	space	261	y space	121	y	y
98	b	262	space b	32	space	space
105	i	263	b i	98	b	b
116	t	-	-	-	-	-
116	t	264	t t t	258	i t	-
121	y	-	-	-	-	-
32	space	265	t y space	260	t y	t y
98	b	-	-	-	-	-
105	i	266	space-b i	262	space b	space b

5) nova\_entrada=i

9) nova\_entrada'=spa b

1) Nova\_entrada = space b (tem no dicionário)  
 2) Nova\_entrada = space b i (não tem)  
 ⇒ 3) Insere no dicionário

4) Manda código de nova\_entrada, sem o caractere final ⇒ space b

6) Nova\_entrada' = t y spa  
 7) Insere nova\_entrada' no dicionário  
 8) Saída = cadeia do código recebido ⇒ spa b

■ Exemplo 1: mensagem “*itty\_bitty\_bit bin*”

Codificação

Transmissão

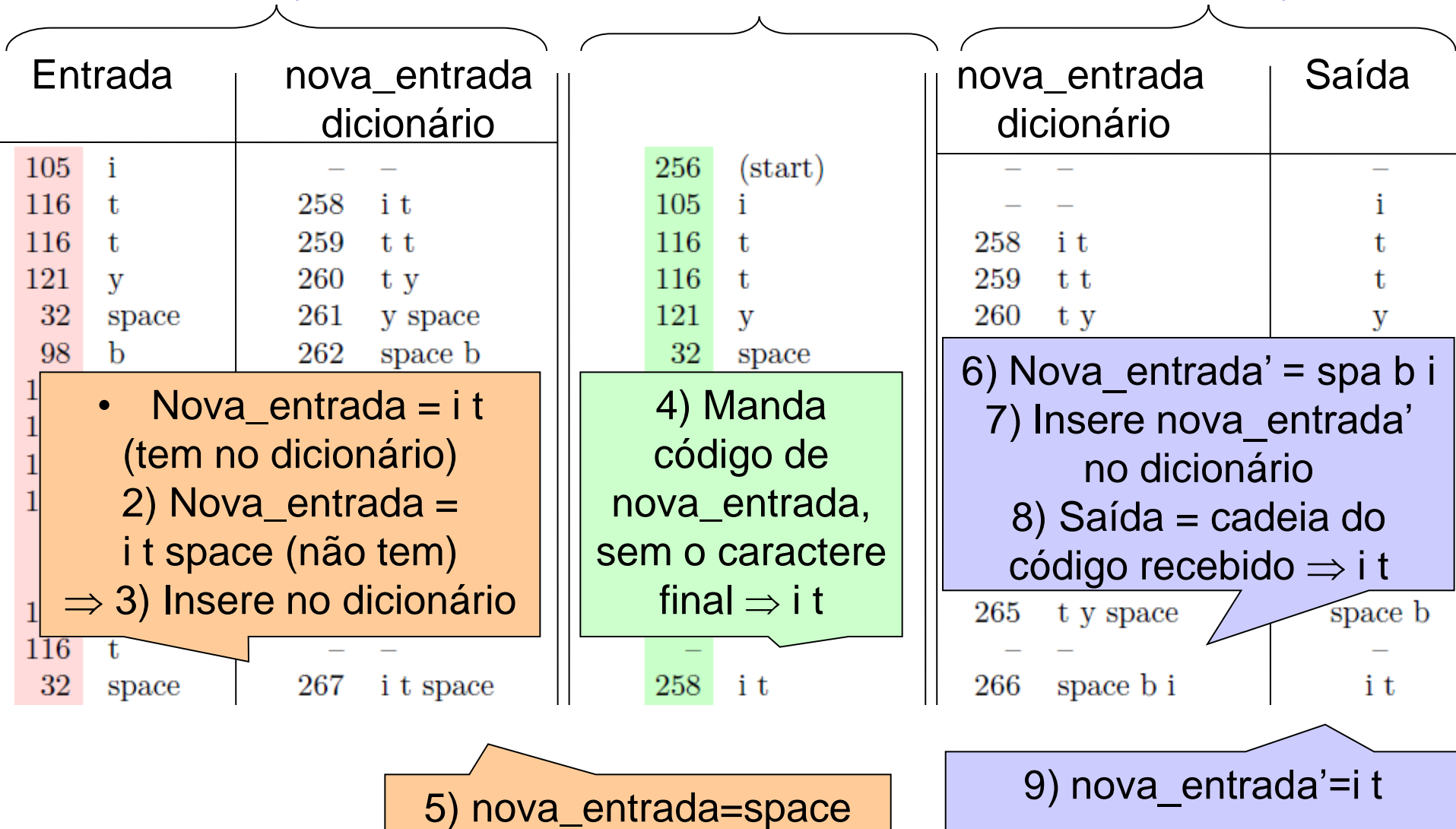
Decodificação

Entrada		nova_entrada dicionário				nova_entrada dicionário		Saída
105	i	—	—	256	(start)	—	—	—
116	t	258	i t	105	i	—	—	i
116	t	259	t t	116	t	258	i t	t
121	y	260	t y	116	t	259	t t	t
32	space	261	y space	121	y	260	t y	y
98	b	262	space b	32	space	261	y space	space
105	i	263	b i	98	b	262	space b	b
116	t	—	—	—	—	—	—	—
116	t	264	i t t	258	i t	263	b i	i t
121	y	—	—	—	—	—	—	—
32	space	265	t y space	260	t y	264	i t t	t y
98	b	—	—	—	—	—	—	—
105	i	266	space-b i	262	space b	265	t y space	space b
116	t	—	—	—	—	—	—	—

- Nova\_entrada = i  
(tem no dicionário)
- 2) Nova\_entrada =  
i t (tem)

# Exemplo 1: mensagem “*itty\_bitty\_bit\_bin*”

Codificação → Transmissão → Decodificação



# Exemplo 1: mensagem “*itty\_bitty\_bit\_bin*”

Codificação

Transmissão

Decodificação

Entrada		nova_entrada dicionário	
105	i	—	—
116	t	258	i t
116	t	259	t t
121	y	260	t y
32	space	261	y space
98	b	262	space b
105	i	263	b i
116	t	—	—
116	t	264	i t t
1			
1			
1			
1			
32	space	267	i t space
98	b	—	—

- Nova\_entrada = space (tem no dicionário)
- 2) Nova\_entrada = space b (tem)

256	(start)
105	i
116	t
116	t
121	y
32	space
98	b
—	—
258	i t
—	—
260	t y
—	—
262	space b
—	—
258	i t
—	—

nova_entrada dicionário		Saída
—	—	—
—	—	i
258	i t	t
259	t t	t
260	t y	y
261	y space	space
262	space b	b
—	—	—
263	b i	i t
—	—	—
264	i t t	t y
—	—	—
265	t y space	space b
—	—	—
266	space b i	i t
—	—	—



# Exemplo 1: mensagem “*itty\_bitty\_bit\_bin*”

Codificação → Transmissão → Decodificação

Entrada		nova_entrada dicionário		nova_entrada dicionário		Saída
105	i	—	—	256	(start)	—
116	t	258	i t	105	i	i
116	t	259	t t	116	t	t
121	y	260	t y	116	t	t
32	space	261	y space	121	y	y
98	b	—	—	32	space	space
105	i	—	—	98	b	b
116	t	266	space-b i	—	—	—
116	t	—	—	258	i t	i t
32	space	267	i t space	—	—	—
98	b	—	—	260	t y	t y
105	i	—	—	—	—	—
				262	space b	space b
				—	—	—
				258	i t	i t
				—	—	—
				—	—	—

- Nova\_entrada = space b (tem)
- 2) Nova\_entrada = space b i (tem)

# Exemplo 1: mensagem “*itty\_bitty\_bit\_bin*”

Codificação

Transmissão

Decodificação

Entrada

nova\_entrada  
dicionário

105	i	—	—
116	t	258	i t
116	t	259	t t
121	y	260	t y
23	space	261	y space

- Nova\_entrada = space b i (tem)
- 2) Nova\_entrada = space b i n (não tem)
- ⇒ 3) Insere no dicionário

98	—	—	—
105	i	266	space-b i
116	t	—	—
32	space	267	i t space
98	b	—	—
105	i	—	—
110	n	268	space b i n

5) nova\_entrada=n

256	(start)
105	i
116	t
116	t
121	y
32	space
98	b
—	—
258	i t

4) Manda código de nova\_entrada, sem o caractere final ⇒ space b i

266	space b i
-----	-----------

nova\_entrada  
dicionário

Saída

—	—	—
—	—	i
258	i t	t
259	t t	t
260	t y	y
261	y space	space
262	space b	b

6) Nova\_entrada' = i t spa  
7) Insere nova\_entrada' no dicionário  
8) Saída = cadeia do código recebido ⇒ spa b i

266	space b i	i t
—	—	—
—	—	—
267	i t space	space b i

9) nova\_entrada'=spa b i

# Exemplo 1: mensagem “*itty\_bitty\_bit\_bin*”

Codificação

Transmissão

Decodificação

Entrada		nova_entrada dicionário	
105	i	—	—
116	t	258	i t
116	t	259	t t
121	y	260	t y
32	space	261	y space
98	b	262	space b
<div> <ul style="list-style-type: none"> <li>Nova_entrada = n (tem)</li> </ul> </div> <div>2) Acabou mensagem</div>			
105	i	266	space-b i
116	t	—	—
32	s	267	i t space
98	b	—	—
105	i	—	—
110	n	268	space b i n
—	—	—	—
—	—	—	—

256	(start)
105	i
116	t
116	t
121	y
32	space
98	b
—	—
<div>3) Manda código de nova_entrada ⇒ n</div> <div>4) Manda parada</div>	
—	—
266	space b i
110	n
257	(stop)

nova_entrada dicionário		Saída
—	—	—
—	—	i
258	i t	t
259	t t	t
260	t y	y
261	y space	space
262	space b	b
<div>6) Nova_entrada' = spa b i n</div> <div>7) Insere nova_entrada' no dicionário</div> <div>8) Saída = cadeia do código recebido ⇒ n</div> <div>9) Recebe parada</div>		
—	—	—
267	i t space	space b i
268	space b i n	n
—	—	—

■ Exemplo 1: mensagem “*itty bitty bit bin*”

Codificação

Transmissão

Decodificação

Entrada		nova_entrada dicionário				nova_entrada dicionário		Saída
105	i	—	—	256	(start)	—	—	—
116	t	258	i t	105	i	—	—	i
116	t	259	t t	116	t	258	i t	t
121	y	260	t y	116	t	259	t t	t
32	space	261	y space	121	y	260	t y	y
98	b	262	space b	32	space	261	y space	space
105	i	263	b i	98	b	262	space b	b
116	t	—	—	—	—	—	—	—
116	t	264	i t t	258	i t	263	b i	i t
121	y	—	—	—	—	—	—	—
32	space	265	t y space	260	t y	264	i t t	t y
98	b	—	—	—	—	—	—	—
105	i	266	space-b i	262	space b	265	t y space	space b
116	t	—	—	—	—	—	—	—
32	space	267	i t space	258	i t	266	space b i	i t
98	b	—	—	—	—	—	—	—
105	i	—	—	—	—	—	—	—
110	n	268	space b i n	266	space b i	267	i t space	space b i
—	—	—	—	110	n	268	space b i n	n
—	—	—	—	257	(stop)	—	—	—

## ■ Exemplo 2: mensagem “aaaaaa”

Codificação → Transmissão → Decodificação

Codificação		Transmissão		Decodificação	
Entrada	nova_entrada dicionário			nova_entrada dicionário	Saída
97 a	- -	256 (início)		- -	-

1) Nova\_entrada =  $\phi$

4) Nova\_entrada = a

2) Manda  
código de início

3) Recebendo  
código de início,  
limpa dicionário e faz  
nova\_entrada' =  $\phi$

■ Exemplo 2: mensagem “**aa**aaaa”

Codificação → Transmissão → Decodificação

Codificação		Transmissão		Decodificação	
Entrada	nova_entrada dicionário			nova_entrada dicionário	Saída
97 a	- -	256 (início)		- -	-
97 a	258 aa	97 (a)		- -	a

1) Nova\_entrada = aa  
(não tem no dicionário)  
⇒ 2) Insere no dicionário

3) Manda  
código de  
nova\_entrada,  
sem o caractere  
final ⇒ a

5) Produz como saída  
o caractere do código  
recebido ⇒ a  
6) Nova\_entrada' = a

4) nova\_entrada=a  
(último caractere  
recebido)

■ Exemplo 2: mensagem “**aaaaaa**”

Codificação → Transmissão → Decodificação

Codificação		Transmissão		Decodificação	
Entrada	nova_entrada dicionário			nova_entrada dicionário	Saída
97 a	- -	256 (início)		- -	-
97 a	258 aa	97 (a)		- -	a
97 a	- -	- -		- -	-

- 1) Nova\_entrada = a  
(tem no dicionário)
- 2) Nova\_entrada = aa  
(tem)

## ■ Exemplo 2: mensagem “aaaaaa”

Codificação → Transmissão → Decodificação

Codificação		Transmissão		Decodificação	
Entrada	nova_entrada dicionário			nova_entrada dicionário	Saída
97 a	- -	256 (início)		- -	-
97 a	258 aa	97 (a)		- -	a
97 a	- -	- -		- -	-
97 a	259 aaa	258 (aa)		258 aa	aa

1) Nova\_entrada = aa  
(tem no dicionário)  
2) Nova\_entrada = aaa  
(não tem)  
⇒ 3) Insere no dicionário

5) nova\_entrada=a (último  
caractere recebido)

4) Manda  
código de  
nova\_entrada,  
sem o caractere  
final ⇒ aa

6) Não conhece código  
258 ⇒ repete 1o  
caractere de cód. anter.  
em Nova\_entrada' = aa  
7) Insere nova\_entrada'  
no dicionário  
8) Produz saída = cadeia  
do código recebido ⇒ aa

9) nova\_entrada'=aa



■ Exemplo 2: mensagem “**aaaaaa**”

Codificação → Transmissão → Decodificação

Codificação		Transmissão		Decodificação	
Entrada	nova_entrada dicionário			nova_entrada dicionário	Saída
97 a	- -	256 (início)		- -	-
97 a	258 aa	97 (a)		- -	a
97 a	- -	- -		- -	-
97 a	259 aaa	258 (aa)		258 aa	aa
97 a	- -	- -		- -	-

- 1) Nova\_entrada = a  
(tem no dicionário)
- 2) Nova\_entrada = aa  
(tem)

■ Exemplo 2: mensagem “**aaaaaa**”

Codificação → Transmissão → Decodificação

Codificação		Transmissão		Decodificação	
Entrada	nova_entrada dicionário			nova_entrada dicionário	Saída
97 a	- -	256 (início)		- -	-
97 a	258 aa	97 (a)		- -	a
97 a	- -	- -		- -	-
97 a	259 aaa	258 (aa)		258 aa	aa
97 a	- -	- -		- -	-
97 a	- -	- -		- -	-

- 1) Nova\_entrada = aa  
(tem no dicionário)
- 2) Nova\_entrada = aaa  
(tem)

## ■ Exemplo 2: mensagem “**aaaaaa**”

Codificação → Transmissão → Decodificação

Entrada	nova_entrada dicionário
97 a	- -
97 a	258 aa
97 a	- -
97 a	259 aaa
97 a	- -
97 a	- -
- -	- -
- -	- -

- 1) Nova\_entrada = aaa  
(tem no dicionário)
- 2) Recebeu parada

256 (início)  
97 (a)  
- -  
258 (aa)  
- -  
- -  
259 (aaa)  
257 (parada)

- 4) Manda  
código de  
nova\_entrada,  
seguido por  
parada

nova_entrada dicionário	Saída
- -	-
- -	a
- -	-
258 aa	aa
- -	-
- -	-
259 aaa	aaa

- 6) Não conhece código 259  
⇒ repete 1o carac. de cód.  
ant. em nova\_entrada' = aaa
- 7) Insere nova\_entrada' no  
dicionário
- 8) Produz saída = cadeia do  
código recebido ⇒ aaa

■ Exemplo 2: mensagem “**aaaaaa**”

Codificação → Transmissão → Decodificação

Codificação		Transmissão		Decodificação	
Entrada	nova_entrada dicionário			nova_entrada dicionário	Saída
97 a	- -	256 (início)		- -	-
97 a	258 aa	97 (a)		- -	a
97 a	- -	- -		- -	-
97 a	259 aaa	258 (aa)		258 aa	aa
97 a	- -	- -		- -	-
97 a	- -	- -		- -	-
- -	- -	259 (aaa)		259 aaa	aaa
- -	- -	257 (parada)		- -	-

# Compressão LZW

- Codificador e decodificador criam dicionário a medida que processamento é feito
  - Dicionário não precisa ser transmitido explicitamente
  - Codificador lida com o texto em um único passo

# Compressão LZW

- O número de bits transmitido é reduzido?
  - **Exemplo 1:**
    - Manda 18 caracteres de 8 bits (144 bits) em 14 transmissões de 9 bits (126 bits)
      - Redução de 12,5%
  - **Exemplo 2:**
    - Manda 6 caracteres de 8 bits (48 bits) em 5 transmissões de 9 bits (45 bits)
      - Redução de 6,25%

# Exercícios

- ❑ Codifique e decodifique a mensagem “***itty bitty nitty grrrity bit bin***”
    - Como feito no exemplo anterior
  - ❑ Qual foi a compactação obtida?
  - ❑ Você notou situações em que poderia haver melhoras no processo?
-

# Discussão: compressão sem perda

- É usada em várias aplicações
    - **Exemplos:** formato zip do Unix, formato GIF
  - É usada em casos em que é importante que os dados originais e descomprimidos sejam idênticos (ou desvios nos dados originais podem ser prejudiciais)
    - **Exemplos:** programas executáveis, programa fonte, texto, registros de banco, artigos etc.
-



# Discussão: compressão sem perda

- Não há algoritmo sem perda eficiente para todos os tipos de dados possíveis
- Sequências de dados completamente aleatórias não podem ser comprimidas