

# Compressão de Dados

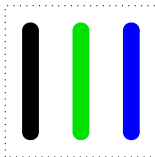
## Algoritmo de Huffman

Prof. Kennedy Reurison Lopes

August 28, 2024

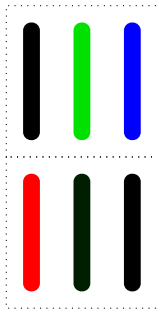
# Codificação

Uma imagem numa tela é composta por uma sequência de pixels no formato RGB.



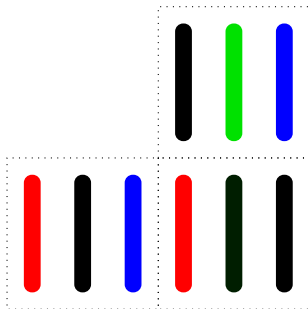
# Codificação

Uma imagem numa tela é composta por uma sequência de pixels no formato RGB.



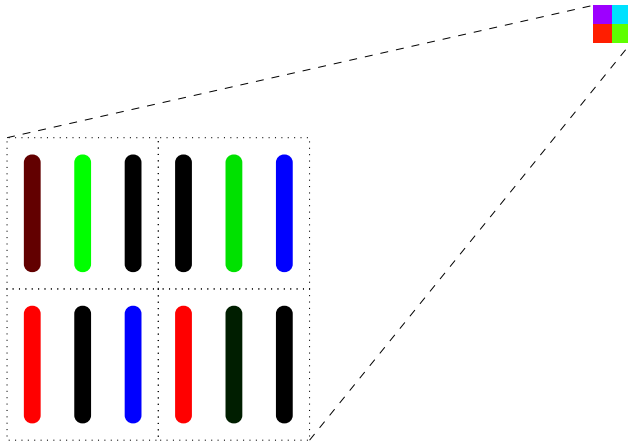
# Codificação

Uma imagem numa tela é composta por uma sequência de pixels no formato RGB.



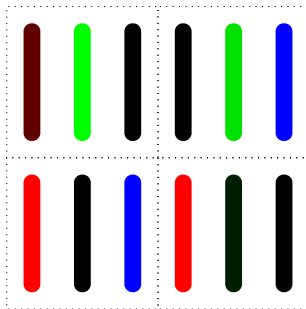
# Codificação

Uma imagem numa tela é composta por uma sequência de pixels no formato RGB.



# Codificação

Uma imagem numa tela é composta por uma sequência de pixels no formato RGB.



# Codificação - Imagem RAW

Considere uma imagem no formato *raw*<sup>1</sup>. O arquivo desta imagem preserva a qualidade original da foto, sem perder nem um *bit* de informação.

Uma imagem em 4K desta imagem tem a seguinte quantidade de pixels:

$$P = 3840 \times 2160 = 8.294.400 \text{ pixels}$$

Cada Pixel contém uma cor no formato RGB(3 bytes):

$$S = 3840 \times 2160 \times 3 = 24.883.200 \text{ bytes}$$

Cada 1.024 bytes corresponde a 1kB e cada 1.024kB corresponde a 1MB. Portanto, uma imagem RAW em formato 4K contém aproximadamente:

$$S \simeq 23.73 \text{ MB}$$

---

<sup>1</sup>Um arquivo RAW contém todos os dados de uma imagem não compactados e não processados capturados por um scanner ou pelos sensores de uma câmera digital. As fotos feitas no formato RAW contêm alto nível de detalhes, são grandes e não têm perdas de qualidade. ↻ 🔍 🔍

# Video 4K

Imagine a situação que os dados são transmitidos pixel a pixel no formato 4K.  
Uma hora deste filme custaria aproximadamente:

---

<sup>2</sup>Mídia digital capaz de armazenar até 25GB de dados



# Video 4K

Imagine a situação que os dados são transmitidos pixel a pixel no formato 4K.  
Uma hora deste filme custaria aproximadamente:

$$S = (1h) \times \left( \frac{23.73 \text{ MB}}{\text{frame}} \right) \times$$

# Video 4K

Imagine a situação que os dados são transmitidos pixel a pixel no formato 4K.  
Uma hora deste filme custaria aproximadamente:

$$S = (1h) \times \left( \frac{23.73 \text{ MB}}{\text{frame}} \right) \times \left( \frac{120 \text{ frame}}{s} \right) \times$$

# Video 4K

Imagine a situação que os dados são transmitidos pixel a pixel no formato 4K.  
Uma hora deste filme custaria aproximadamente:

$$S = (1h) \times \left( \frac{23.73 \text{ MB}}{\text{frame}} \right) \times \left( \frac{120 \text{ frame}}{s} \right) \times \left( \frac{60 \text{ min}}{1h} \right) \times$$

# Video 4K

Imagine a situação que os dados são transmitidos pixel a pixel no formato 4K.  
Uma hora deste filme custaria aproximadamente:

$$S = (1h) \times \left( \frac{23.73 \text{ MB}}{\text{frame}} \right) \times \left( \frac{120 \text{ frame}}{s} \right) \times \left( \frac{60 \text{ min}}{1h} \right) \times \left( \frac{60s}{1 \text{ min}} \right)$$

# Video 4K

Imagine a situação que os dados são transmitidos pixel a pixel no formato 4K.  
Uma hora deste filme custaria aproximadamente:

$$S = (1h) \times \left( \frac{23.73 \text{ MB}}{\text{frame}} \right) \times \left( \frac{120 \text{ frame}}{s} \right) \times \left( \frac{60 \text{ min}}{1h} \right) \times \left( \frac{60s}{1\text{min}} \right)$$

$$S = 23.73 \times 120 \times 60 \times 60 \times \frac{\text{MB} \times h \times \text{frame} \times \text{min} \times s}{\text{frame} \times s \times h \times \text{min}}$$

# Video 4K

Imagine a situação que os dados são transmitidos pixel a pixel no formato 4K.  
Uma hora deste filme custaria aproximadamente:

$$S = (1h) \times \left( \frac{23.73 \text{ MB}}{\text{frame}} \right) \times \left( \frac{120 \text{ frame}}{s} \right) \times \left( \frac{60 \text{ min}}{1h} \right) \times \left( \frac{60s}{1 \text{ min}} \right)$$

$$S = 23.73 \times 120 \times 60 \times 60 \times \frac{\text{MB} \times h \times \text{frame} \times \text{min} \times s}{\text{frame} \times s \times h \times \text{min}}$$

$$S = 1.025.136.000 \text{ MB} \simeq 1.001.110 \text{ GB} \simeq 977 \text{ TB}$$

# Video 4K

Imagine a situação que os dados são transmitidos pixel a pixel no formato 4K. Uma hora deste filme custaria aproximadamente:

$$S = (1h) \times \left( \frac{23.73 \text{ MB}}{\text{frame}} \right) \times \left( \frac{120 \text{ frame}}{s} \right) \times \left( \frac{60 \text{ min}}{1h} \right) \times \left( \frac{60s}{1 \text{ min}} \right)$$

$$S = 23.73 \times 120 \times 60 \times 60 \times \frac{\text{MB} \times h \times \text{frame} \times \text{min} \times s}{\text{frame} \times s \times h \times \text{min}}$$

$$S = 1.025.136.000 \text{ MB} \simeq 1.001.110 \text{ GB} \simeq 977 \text{ TB}$$

Obviamente os filmes não ocupam este espaço no disco. Então como é possível armazenar um filme em um Blue-Ray<sup>2</sup>, ou mesmo transferir o filme em 4K no *streaming*?

---

<sup>2</sup>Mídia digital capaz de armazenar até 25GB de dados

# Video 4K

Imagine a situação que os dados são transmitidos pixel a pixel no formato 4K. Uma hora deste filme custaria aproximadamente:

$$S = (1h) \times \left( \frac{23.73 \text{ MB}}{\text{frame}} \right) \times \left( \frac{120 \text{ frame}}{s} \right) \times \left( \frac{60 \text{ min}}{1h} \right) \times \left( \frac{60s}{1 \text{ min}} \right)$$

$$S = 23.73 \times 120 \times 60 \times 60 \times \frac{\text{MB} \times h \times \text{frame} \times \text{min} \times s}{\text{frame} \times s \times h \times \text{min}}$$

$$S = 1.025.136.000 \text{ MB} \simeq 1.001.110 \text{ GB} \simeq 977 \text{ TB}$$

Obviamente os filmes não ocupam este espaço no disco. Então como é possível armazenar um filme em um Blue-Ray<sup>2</sup>, ou mesmo transferir o filme em 4K no *streaming*?

Resposta: Compactação.

---

<sup>2</sup>Mídia digital capaz de armazenar até 25GB de dados



# Codificação - Imagem RAW

Algumas considerações:

# Codificação - Imagem RAW

Algumas considerações:

Como é um byte para cada cor, o computador pode gerar até:

$$(2^8)^3 = 2^{24} = 16.777.216 \text{ cores}$$

# Codificação - Imagem RAW

Algumas considerações:

Como é um byte para cada cor, o computador pode gerar até:

$$(2^8)^3 = 2^{24} = 16.777.216 \text{ cores}$$

O ser humano consegue enxergar *apenas* 1 milhão de cores.

# Codificação - Imagem RAW

Algumas considerações:

Como é um byte para cada cor, o computador pode gerar até:

$$(2^8)^3 = 2^{24} = 16.777.216 \text{ cores}$$

O ser humano consegue enxergar *apenas* 1 milhão de cores.

Em uma imagem, não são utilizadas todas estas cores. Exemplos:

# Codificação - Imagem RAW

Algumas considerações:

Como é um byte para cada cor, o computador pode gerar até:

$$(2^8)^3 = 2^{24} = 16.777.216 \text{ cores}$$

O ser humano consegue enxergar *apenas* 1 milhão de cores.

Em uma imagem, não são utilizadas todas estas cores. Exemplos:

Uma imagem  $800 \times 600$  precisa ter no máximo 480.000 cores;

# Codificação - Imagem RAW

Algumas considerações:

Como é um byte para cada cor, o computador pode gerar até:

$$(2^8)^3 = 2^{24} = 16.777.216 \text{ cores}$$

O ser humano consegue enxergar *apenas* 1 milhão de cores.

Em uma imagem, não são utilizadas todas estas cores. Exemplos:

Uma imagem  $800 \times 600$  precisa ter no máximo 480.000 cores;

A resolução HD precisa de no máximo  $1280 \times 720 = 921.600$  cores.

# Codificação - Imagem RAW

Algumas considerações:

Como é um byte para cada cor, o computador pode gerar até:

$$(2^8)^3 = 2^{24} = 16.777.216 \text{ cores}$$

O ser humano consegue enxergar *apenas* 1 milhão de cores.

Em uma imagem, não são utilizadas todas estas cores. Exemplos:

Uma imagem  $800 \times 600$  precisa ter no máximo 480.000 cores;

A resolução HD precisa de no máximo  $1280 \times 720 = 921.600$  cores.

A própria resolução 4K precisa apenas de 8.294.400 cores.

# Codificação - Imagem RAW

Algumas considerações:

Como é um byte para cada cor, o computador pode gerar até:

$$(2^8)^3 = 2^{24} = 16.777.216 \text{ cores}$$

O ser humano consegue enxergar *apenas* 1 milhão de cores.

Em uma imagem, não são utilizadas todas estas cores. Exemplos:

Uma imagem  $800 \times 600$  precisa ter no máximo 480.000 cores;

A resolução HD precisa de no máximo  $1280 \times 720 = 921.600$  cores.

A própria resolução 4K precisa apenas de 8.294.400 cores.

Existem muitos pixels repetidos (cores repetidas).



# Codificação de texto

Um texto é composto por uma sequência de caracteres (imprimíveis e não-imprimíveis) sendo, cada caractere representado por uma sequência de bits.  
Exemplo:

*K – E – N – N – E – D – Y*

Utilizando a codificação ASCII pode ser:

75 – 101 – 110 – 110 – 101 – 100 – 121

Sendo que cada símbolo deste é representado por uma sequência binária com 8 bits.

Cada caractere possui exatamente 1byte de representação. Desta forma, o meu nome ocupa:

56 bits

# Codificação de texto

Vamos considerar um texto maior: **Um prato de trigo para tres tigres tristes:**

*U m \_ p r a t o*

85 – 109 – 95 – 112 – 114 – 97 – 116 – 111

*d e \_ t r i g o*

100 – 101 – 95 – 116 – 114 – 105 – 103 – 111

*p a r a \_ t r e s*

112 – 97 – 114 – 97 – 95 – 116 – 114 – 101 – 115

*t i g r e s \_ t r i s t e s*

116 – 105 – 103 – 114 – 101 – 115 – 95 – 116 – 114 – 105 – 115 – 116 – 101 – 115

# Codificação de texto

Vamos considerar um texto maior: **Um prato de trigo para tres tigres tristes:**  
Este texto contem:

# Codificação de texto

Vamos considerar um texto maior: **Um prato de trigo para tres tigres tristes:**  
Este texto contem:

42 caracteres.

# Codificação de texto

Vamos considerar um texto maior: **Um prato de trigo para tres tigres tristes:**  
Este texto contem:

42 caracteres.

13 caracteres distintos.

# Codificação de texto

Vamos considerar um texto maior: **Um prato de trigo para tres tigres tristes:**  
Este texto contem:

42 caracteres.

13 caracteres distintos.

O texto ocupa 336 bits.

# Codificação de texto

Vamos considerar um texto maior: **Um prato de trigo para tres tigres tristes:**

Este texto contem:

42 caracteres.

13 caracteres distintos.

O texto ocupa 336 bits.

É possível ocupar menos espaço?

# Codificação de texto

Vamos considerar um texto maior: **Um prato de trigo para tres tigres tristes:**

Este texto contem:

42 caracteres.

13 caracteres distintos.

O texto ocupa 336 bits.

É possível ocupar menos espaço?

Vamos analisar a frequência dos caracteres:



# Análise de Frequência

Frequência dos caracteres:

# Análise de Frequência

Frequência dos caracteres:

U	m	-	p	r	a	t	o	d	e	i	g	s
1	1	7	2	6	3	6	2	1	4	3	2	4

Ordenando as frequências:

# Análise de Frequência

Frequência dos caracteres:

U	m	-	p	r	a	t	o	d	e	i	g	s
1	1	7	2	6	3	6	2	1	4	3	2	4

Ordenando as frequências:

U	m	d	p	o	g	a	i	e	s	r	t	-
1	1	1	2	2	2	3	3	4	4	6	6	7

Agrupando as menores frequências:

# Análise de Frequência

Frequência dos caracteres:

U	m	-	p	r	a	t	o	d	e	i	g	s
1	1	7	2	6	3	6	2	1	4	3	2	4

Ordenando as frequências:

U	m	d	p	o	g	a	i	e	s	r	t	-
1	1	1	2	2	2	3	3	4	4	6	6	7

Agrupando as menores frequências:

U	$\{d, m\}$	p	o	g	a	i	e	s	r	t	-
1	2	2	2	2	3	3	4	4	6	6	7

# Análise de Frequência

O algoritmo continua (agrupando os menores) até não for mais possível continuar:

Agrupando( $U$  e  $\{d, m\}$ ):

# Análise de Frequência

O algoritmo continua (agrupando os menores) até não for mais possível continuar:

Agrupando( $U$  e  $\{d, m\}$ ):

$\{U, d, m\}$	p	o	g	a	i	e	s	r	t	-
3	2	2	2	3	3	4	4	6	6	7

Ordenando:

# Análise de Frequência

O algoritmo continua (agrupando os menores) até não for mais possível continuar:

Agrupando( $U$  e  $\{d, m\}$ ):

$\{U, d, m\}$	p	o	g	a	i	e	s	r	t	-
3	2	2	2	3	3	4	4	6	6	7

Ordenando:

g	p	o	$\{U, d, m\}$	a	i	e	s	r	t	-
2	2	2	3	3	3	4	4	6	6	7

# Análise de Frequência



# Análise de Frequência

g	$\{p, o\}$	$\{U, d, m\}$	a	i	e	s	r	t	-
2	4	3	3	3	4	4	6	6	7

# Análise de Frequência

g	{p, o}	{U, d, m}	a	i	e	s	r	t	-
2	4	3	3	3	4	4	6	6	7

{g, a}	{p, o}	{U, d, m}	i	e	s	r	t	-
5	4	3	3	4	4	6	6	7

# Análise de Frequência

g	{p, o}	{U, d, m}	a	i	e	s	r	t	-
2	4	3	3	3	4	4	6	6	7

{g, a}	{p, o}	{U, d, m}	i	e	s	r	t	-
5	4	3	3	4	4	6	6	7

{g, a}	{p, o}	{U, d, m, i}	e	s	r	t	-
5	4	6	4	4	6	6	7

# Análise de Frequência

g	{p, o}	{U, d, m}	a	i	e	s	r	t	-
2	4	3	3	3	4	4	6	6	7

{g, a}	{p, o}	{U, d, m}	i	e	s	r	t	-
5	4	3	3	4	4	6	6	7

{g, a}	{p, o}	{U, d, m, i}	e	s	r	t	-
5	4	6	4	4	6	6	7

{g, a}	{p, o}	{U, d, m, i}	{e, s}	r	t	-
5	4	6	8	6	6	7

# Análise de Frequência

# Análise de Frequência

$\{g, a, p, o\}$	$\{U, d, m, i\}$	$\{e, s\}$	r	t	-
9	6	8	6	6	7

# Análise de Frequência

$\{g, a, p, o\}$	$\{U, d, m, i\}$	$\{e, s\}$	r	t	-
9	6	8	6	6	7

$\{g, a, p, o\}$	$\{U, d, m, i\}$	$\{e, s\}$	$\{r, t\}$	-
9	6	8	12	7

# Análise de Frequência

$\{g, a, p, o\}$	$\{U, d, m, i\}$	$\{e, s\}$	r	t	-
9	6	8	6	6	7

$\{g, a, p, o\}$	$\{U, d, m, i\}$	$\{e, s\}$	$\{r, t\}$	-
9	6	8	12	7

$\{g, a, p, o\}$	$\{U, d, m, i, -\}$	$\{e, s\}$	$\{r, t\}$
9	13	8	12



# Análise de Frequência

$\{g, a, p, o\}$	$\{U, d, m, i\}$	$\{e, s\}$	$r$	$t$	$-$
9	6	8	6	6	7

$\{g, a, p, o\}$	$\{U, d, m, i\}$	$\{e, s\}$	$\{r, t\}$	$-$
9	6	8	12	7

$\{g, a, p, o\}$	$\{U, d, m, i, -\}$	$\{e, s\}$	$\{r, t\}$
9	13	8	12

$\{g, a, p, o, e, s\}$	$\{U, d, m, i, -\}$	$\{r, t\}$
17	13	12

# Análise de Frequência

# Análise de Frequência

$\{g, a, p, o, e, s\}$	$\{U, d, m, i, -, r, t\}$
17	25

# Análise de Frequência

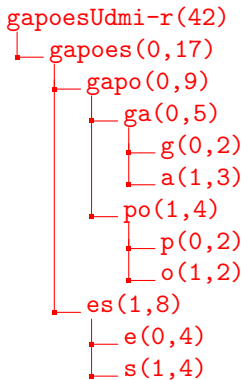
$$\frac{\{g, a, p, o, e, s\}}{17} \quad | \quad \frac{\{U, d, m, i, -, r, t\}}{25}$$

$$\frac{\{g, a, p, o, e, s, U, d, m, i, -, r, t\}}{42}$$

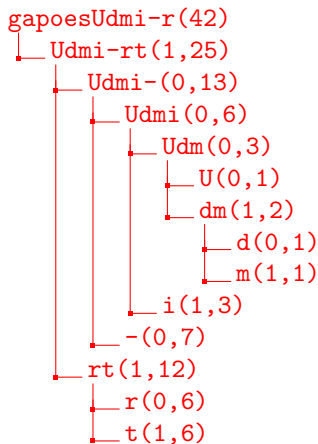
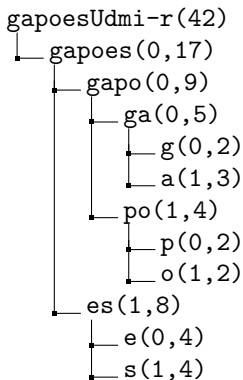
# Construção da árvore

# Construção da árvore

# Construção da árvore

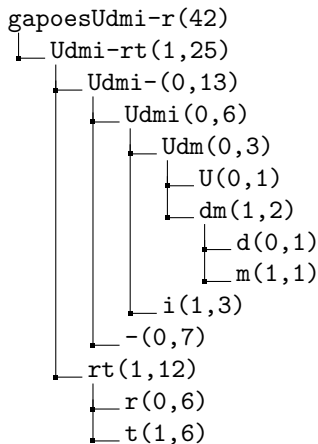
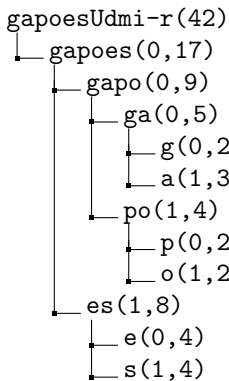


# Construção da árvore

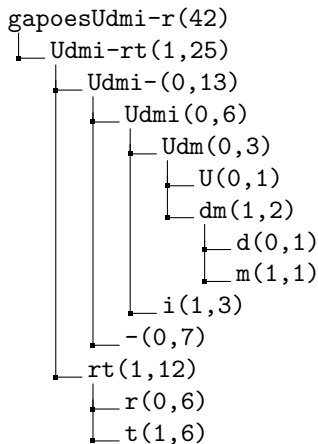
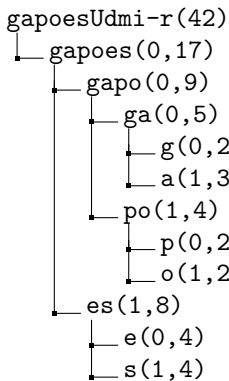




# Construção da árvore

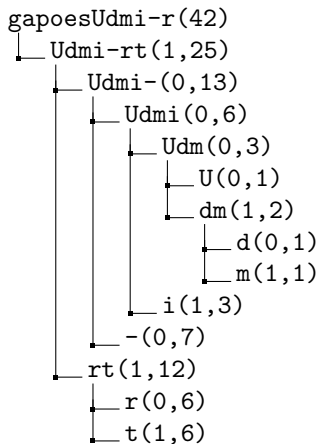
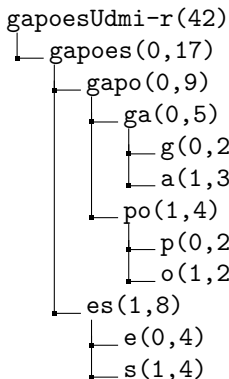


# Construção da árvore



U → 10000

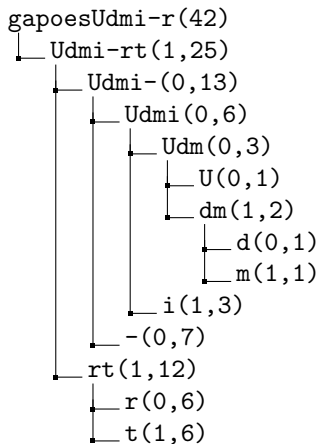
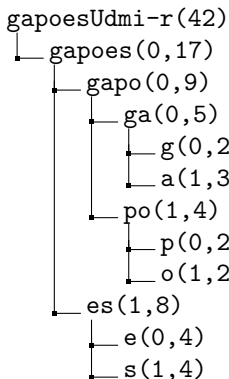
# Construção da árvore



U → 10000

m → 1000011

# Construção da árvore

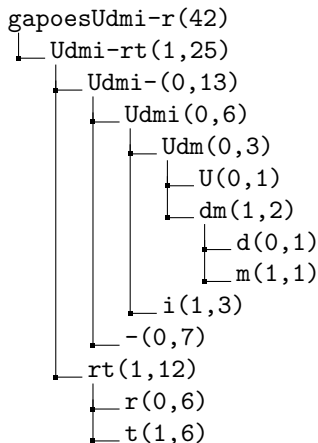
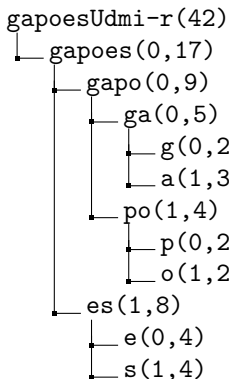


U → 10000

m → 1000011

d → 1000010

# Construção da árvore



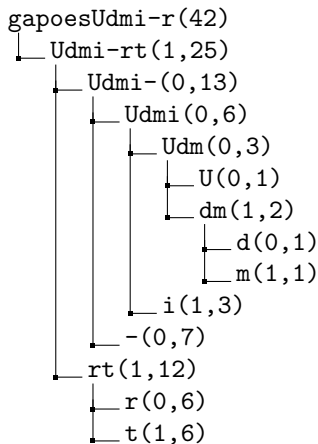
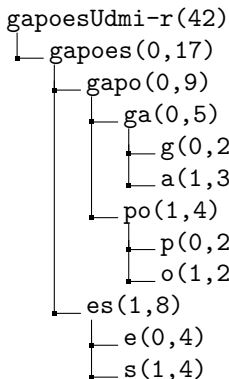
U → 10000

m → 1000011

d → 1000010

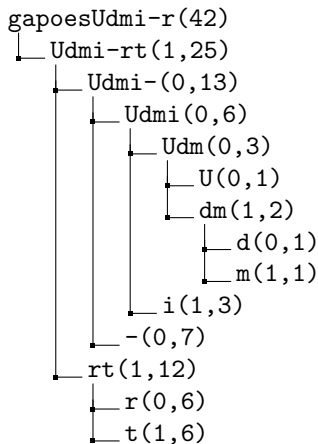
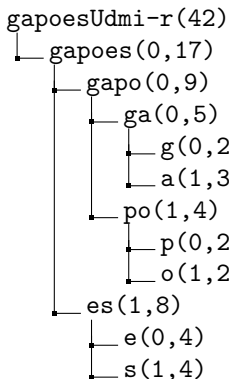
p → 0010

# Construção da árvore



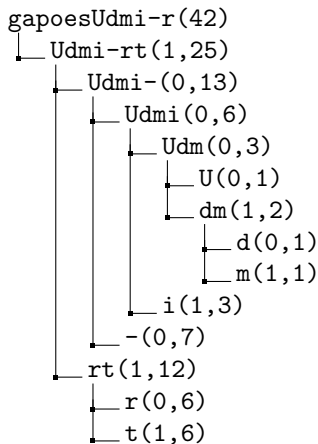
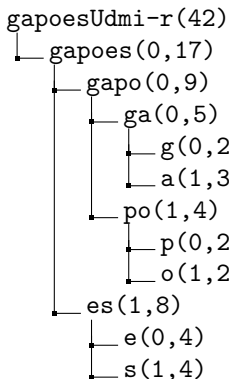
U → 10000  
m → 1000011  
d → 1000010  
p → 0010  
o → 0011

# Construção da árvore



U → 10000  
m → 1000011  
d → 1000010  
p → 0010  
o → 0011  
g → 0000

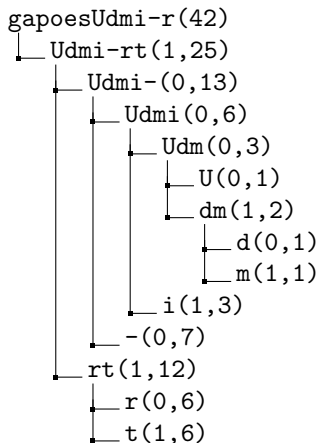
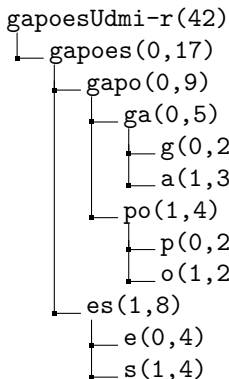
# Construção da árvore



U → 10000  
m → 1000011  
d → 1000010  
p → 0010  
o → 0011  
g → 0000  
a → 0001

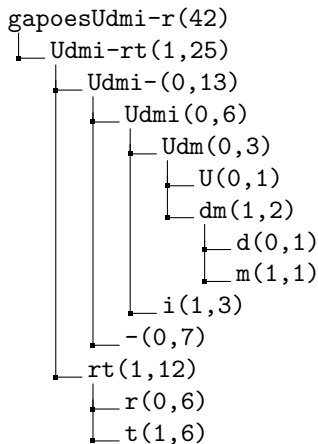
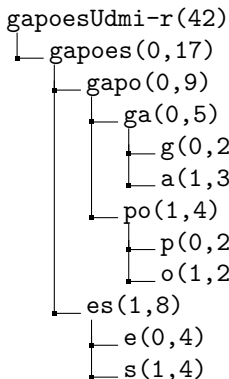


# Construção da árvore



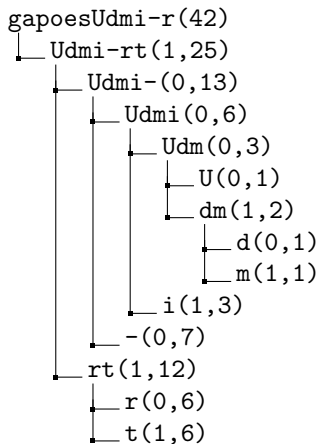
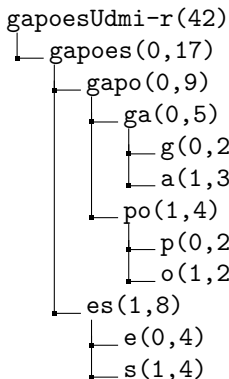
U → 10000  
m → 1000011  
d → 1000010  
p → 0010  
o → 0011  
g → 0000  
a → 0001  
i → 1001

# Construção da árvore



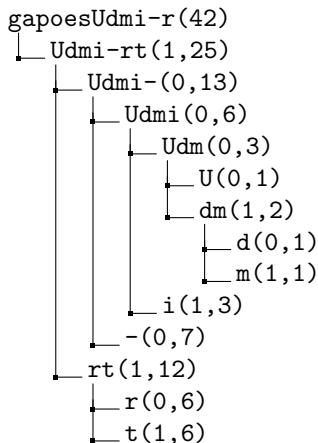
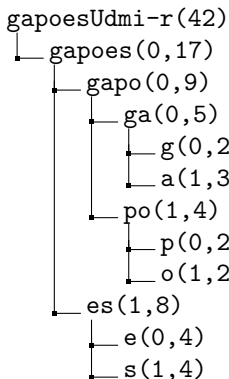
U → 10000  
m → 1000011  
d → 1000010  
p → 0010  
o → 0011  
g → 0000  
a → 0001  
i → 1001  
e → 010

# Construção da árvore



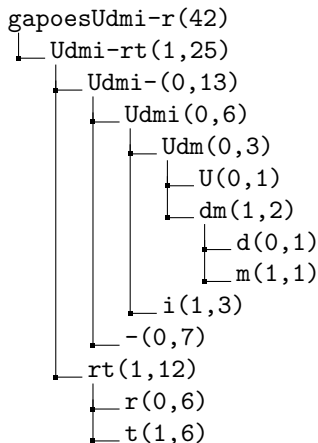
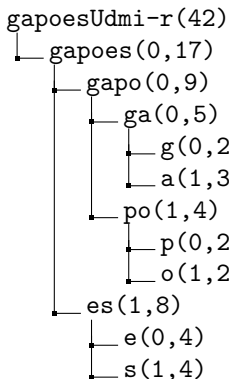
U → 10000  
m → 1000011  
d → 1000010  
p → 0010  
o → 0011  
g → 0000  
a → 0001  
i → 1001  
e → 010  
s → 011

# Construção da árvore



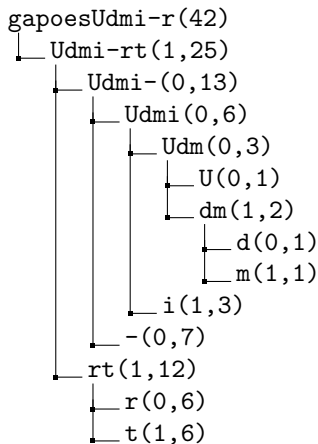
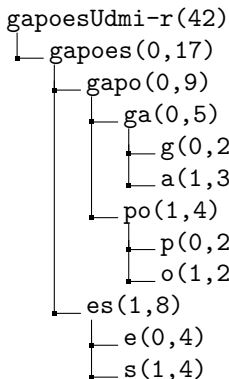
U → 10000  
m → 1000011  
d → 1000010  
p → 0010  
o → 0011  
g → 0000  
a → 0001  
i → 1001  
e → 010  
s → 011  
r → 110

# Construção da árvore



U → 10000  
m → 1000011  
d → 1000010  
p → 0010  
o → 0011  
g → 0000  
a → 0001  
i → 1001  
e → 010  
s → 011  
r → 110  
t → 110

# Construção da árvore



U → 10000  
m → 1000011  
d → 1000010  
p → 0010  
o → 0011  
g → 0000  
a → 0001  
i → 1001  
e → 010  
s → 011  
r → 110  
t → 110  
- → 100

# Análise do armazenamento

char	bin	bits	freq	total
------	-----	------	------	-------

# Análise do armazenamento

char	bin	bits	freq	total
U	10000	5	1	5



# Análise do armazenamento

char	bin	bits	freq	total
U	10000	5	1	5
d	1000010	7	1	7

# Análise do armazenamento

char	bin	bits	freq	total
U	10000	5	1	5
d	1000010	7	1	7
m	1000011	7	1	7

# Análise do armazenamento

char	bin	bits	freq	total
U	10000	5	1	5
d	1000010	7	1	7
m	1000011	7	1	7
p	0010	4	2	8

# Análise do armazenamento

char	bin	bits	freq	total
U	10000	5	1	5
d	1000010	7	1	7
m	1000011	7	1	7
p	0010	4	2	8
o	0011	4	2	8

# Análise do armazenamento

char	bin	bits	freq	total
U	10000	5	1	5
d	1000010	7	1	7
m	1000011	7	1	7
p	0010	4	2	8
o	0011	4	2	8
g	0000	4	2	8

# Análise do armazenamento

char	bin	bits	freq	total
U	10000	5	1	5
d	1000010	7	1	7
m	1000011	7	1	7
p	0010	4	2	8
o	0011	4	2	8
g	0000	4	2	8
a	0001	4	3	12

# Análise do armazenamento

char	bin	bits	freq	total
U	10000	5	1	5
d	1000010	7	1	7
m	1000011	7	1	7
p	0010	4	2	8
o	0011	4	2	8
g	0000	4	2	8
a	0001	4	3	12
i	1001	4	3	12

# Análise do armazenamento

char	bin	bits	freq	total
U	10000	5	1	5
d	1000010	7	1	7
m	1000011	7	1	7
p	0010	4	2	8
o	0011	4	2	8
g	0000	4	2	8
a	0001	4	3	12
i	1001	4	3	12
e	100010	6	4	24



# Análise do armazenamento

char	bin	bits	freq	total
U	10000	5	1	5
d	1000010	7	1	7
m	1000011	7	1	7
p	0010	4	2	8
o	0011	4	2	8
g	0000	4	2	8
a	0001	4	3	12
i	1001	4	3	12
e	100010	6	4	24
s	011	3	4	12

# Análise do armazenamento

char	bin	bits	freq	total
U	10000	5	1	5
d	1000010	7	1	7
m	1000011	7	1	7
p	0010	4	2	8
o	0011	4	2	8
g	0000	4	2	8
a	0001	4	3	12
i	1001	4	3	12
e	100010	6	4	24
s	011	3	4	12
r	100	3	6	18

# Análise do armazenamento

char	bin	bits	freq	total
U	10000	5	1	5
d	1000010	7	1	7
m	1000011	7	1	7
p	0010	4	2	8
o	0011	4	2	8
g	0000	4	2	8
a	0001	4	3	12
i	1001	4	3	12
e	100010	6	4	24
s	011	3	4	12
r	100	3	6	18
t	111	3	6	18

# Análise do armazenamento

char	bin	bits	freq	total
U	10000	5	1	5
d	1000010	7	1	7
m	1000011	7	1	7
p	0010	4	2	8
o	0011	4	2	8
g	0000	4	2	8
a	0001	4	3	12
i	1001	4	3	12
e	100010	6	4	24
s	011	3	4	12
r	100	3	6	18
t	111	3	6	18
-	100	3	7	21

# Análise do armazenamento

char	bin	bits	freq	total
U	10000	5	1	5
d	1000010	7	1	7
m	1000011	7	1	7
p	0010	4	2	8
o	0011	4	2	8
g	0000	4	2	8
a	0001	4	3	12
i	1001	4	3	12
e	100010	6	4	24
s	011	3	4	12
r	100	3	6	18
t	111	3	6	18
-	100	3	7	21
Comprimido				160

# Análise do armazenamento

char	bin	bits	freq	total
U	10000	5	1	5
d	1000010	7	1	7
m	1000011	7	1	7
p	0010	4	2	8
o	0011	4	2	8
g	0000	4	2	8
a	0001	4	3	12
i	1001	4	3	12
e	100010	6	4	24
s	011	3	4	12
r	100	3	6	18
t	111	3	6	18
-	100	3	7	21
Comprimido				160
Comp. médio				3.81

# Análise do armazenamento

Antes do armazenamento:

$$S = 42 \times 8 = 336 \text{ bits}$$

# Análise do armazenamento

Antes do armazenamento:

$$S = 42 \times 8 = 336 \text{ bits}$$

Depois do armazenamento:

$$s = 5 + 7 + 7 + 8 + 8 + 8 + 12 + 12 + 24 + 12 + 18 + 21 = 160 \text{ bits}$$



# Análise do armazenamento

Antes do armazenamento:

$$S = 42 \times 8 = 336 \text{ bits}$$

Depois do armazenamento:

$$s = 5 + 7 + 7 + 8 + 8 + 8 + 12 + 12 + 24 + 12 + 18 + 21 = 160 \text{ bits}$$

Tabela de Símbolos:

$$t = 13 \times 8 = 104 \text{ bits}$$

# Análise do armazenamento

Antes do armazenamento:

$$S = 42 \times 8 = 336 \text{ bits}$$

Depois do armazenamento:

$$s = 5 + 7 + 7 + 8 + 8 + 8 + 12 + 12 + 24 + 12 + 18 + 21 = 160 \text{ bits}$$

Tabela de Símbolos:

$$t = 13 \times 8 = 104 \text{ bits}$$

Compactado:

$$c = S - (s + t) = 336 - (160 + 104) = 72 \text{ bits}$$

# Análise do armazenamento

Antes do armazenamento:

$$S = 42 \times 8 = 336 \text{ bits}$$

Depois do armazenamento:

$$s = 5 + 7 + 7 + 8 + 8 + 8 + 12 + 12 + 24 + 12 + 18 + 21 = 160 \text{ bits}$$

Tabela de Símbolos:

$$t = 13 \times 8 = 104 \text{ bits}$$

Compactado:

$$c = S - (s + t) = 336 - (160 + 104) = 72 \text{ bits}$$

Taxa de compressão:

$$T_x = \frac{c}{T} = \frac{72}{336} = 21.42\%$$

# Resultados

Antes da compressão(em binário):

```
010101010110110101011110111000001110010011000010111010001101111
0101111101100100011001010101111101110100011100100110100101100111
0110111101011111011100000110000101110010011000010101111101110100
0111001001100101011100110101111101110100011010010110011101110010
0110010101110011010111110111010001110010011010010111001101110100
0110010101110011
```

Após a compressão:

```
1000010000111000010110000111000111001000010010100110110100100000
0111000010000111000011001101100100111001101001000011001001110011
0110100101111001001101010101011011010101111101110000011100100110
00010111010001101111
```

# Exercícios

# Exercícios

- ① Calcule a taxa de compressão de: "ABABABACBABABABA".

# Exercícios

- ① Calcule a taxa de compressão de: "ABABABACBABABABA".
- ② Utilize o resultado da compressão dos dados para representar:  
"ACCCCCCCCC".

# Exercícios

- ① Calcule a taxa de compressão de: "ABABABACBABABABA".
- ② Utilize o resultado da compressão dos dados para representar: "ACCCCCCCCC".
- ③ Dado um texto qualquer, a compressão é única? Explique o porquê.



# Exercícios

- ① Calcule a taxa de compressão de: "ABABABACBABABABA".
- ② Utilize o resultado da compressão dos dados para representar: "ACCCCCCCCC".
- ③ Dado um texto qualquer, a compressão é única? Explique o porquê.
- ④ Construa a árvore de Huffman a partir da frequência relativa das letras da língua portuguesa.

\$	f (%)	\$	f (%)	\$	f (%)	\$	f (%)	\$	f (%)
A	14.31	N	5.86	L	2.78	F	0.97	W	0.01
E	12.46	M	4.99	P	2.52	H	0.74	Y	0.01
O	10.73	U	4.86	V	1.52	Z	0.55		
I	8.99	T	4.36	G	1.25	J	0.53		
S	7.07	D	3.53	Q	1.20	X	0.48		
R	6.54	C	3.13	B	1.04	K	0.02		

# Exercícios

# Exercícios

- ⑤ De acordo com a árvore construída na questão anterior, determine a codificação de: 'ABABABACBABABABA'. Apresente a tabela de símbolos.

# Exercícios

- ⑤ De acordo com a árvore construída na questão anterior, determine a codificação de: 'ABABABACBABABABA'. Apresente a tabela de símbolos.
- ⑥ Uma palavra foi codificada usando o código de Huffman, tendo-se obtido a sequência binária: 10111011010111001110100

Caractere	Probabilidade
$P(A)$	0,26
$P(B)$	0,09
$P(C)$	0,08
$P(D)$	0,01
$P(E)$	0,07
$P(I)$	0,22
$P(L)$	0,01
$P(R)$	0,23
$P(T)$	0,03

Sabe-se que o *I* é codificado como sendo 00. Qual é a palavra?