

# Aula 03 – Fundamentos da imagem digital II

Prof. João Fernando Mari

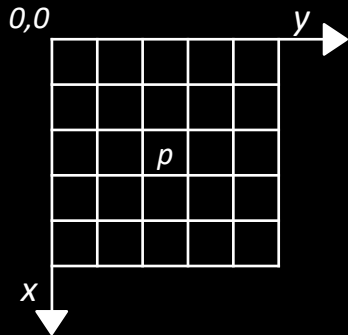
[joaofmari.github.io](https://joaofmari.github.io)

[joaof.mari@ufv.br](mailto:joaof.mari@ufv.br)

- Relacionamento básico entre pixels
  - Vizinhança entre pixels
  - Adjacência
  - Caminho (ou curva) digital
  - Regiões conectadas e componentes conectados
  - Fundo e objetos de uma imagem
  - Borda contorno, ou fronteira
- Operações lógicas e aritméticas entre imagens
  - Operações aritméticas
  - Operações lógicas
- Medidas de distância

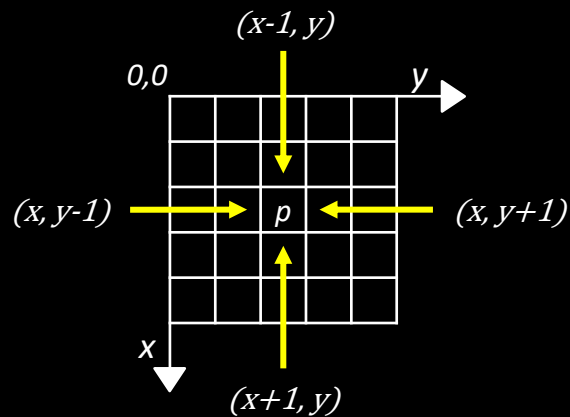
# RELACIONAMENTO BÁSICO ENTRE PIXELS

Vizinhança-4 de  $p$ ,  $N_4(p)$ :



# Vizinhança de um pixel

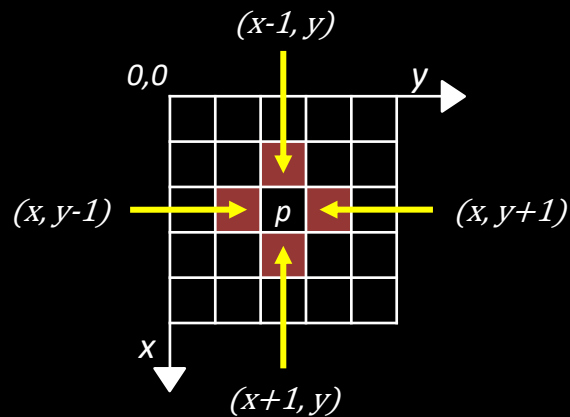
Vizinhança-4 de  $p$ ,  $N_4(p)$ :



# Vizinhança de um pixel

Vizinhança-4 de  $p$ ,  $N_4(p)$ :

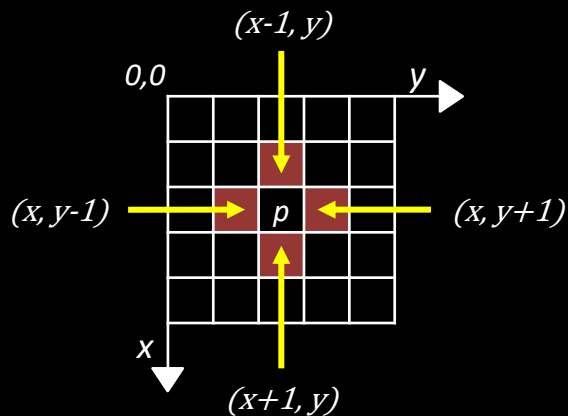
$(x+1, y)$ ,  $(x-1, y)$ ,  $(x, y+1)$ ,  $(x, y-1)$



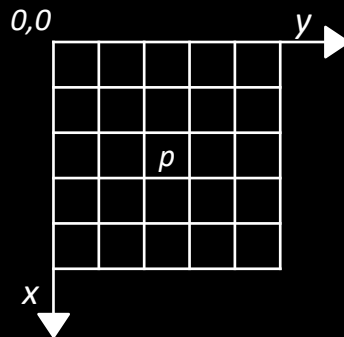
# Vizinhança de um pixel

Vizinhança-4 de  $p$ ,  $N_4(p)$ :

$(x+1, y)$ ,  $(x-1, y)$ ,  $(x, y+1)$ ,  $(x, y-1)$



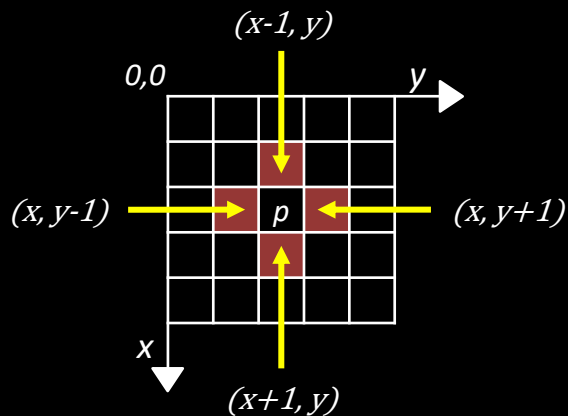
Vizinhança-diagonal de  $p$ ,  $N_D(p)$ :



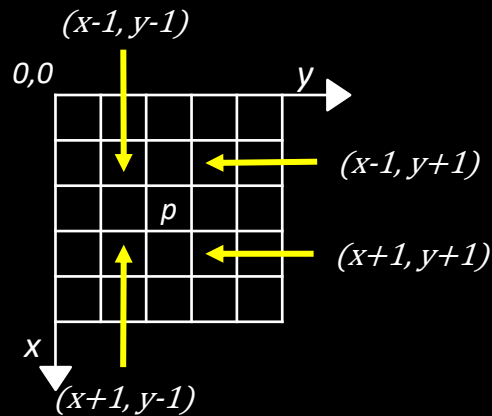
# Vizinhança de um pixel

Vizinhança-4 de  $p$ ,  $N_4(p)$ :

$(x+1, y)$ ,  $(x-1, y)$ ,  $(x, y+1)$ ,  $(x, y-1)$



Vizinhança-diagonal de  $p$ ,  $N_D(p)$ :

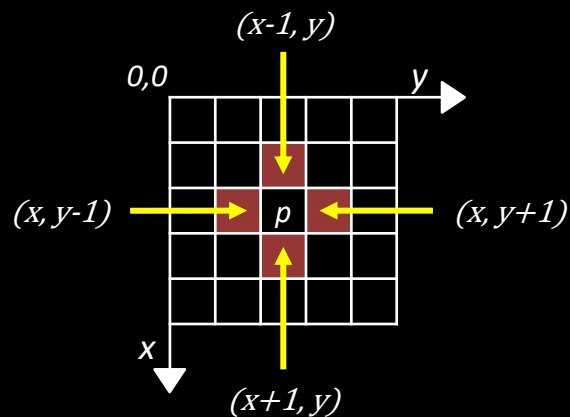




# Vizinhança de um pixel

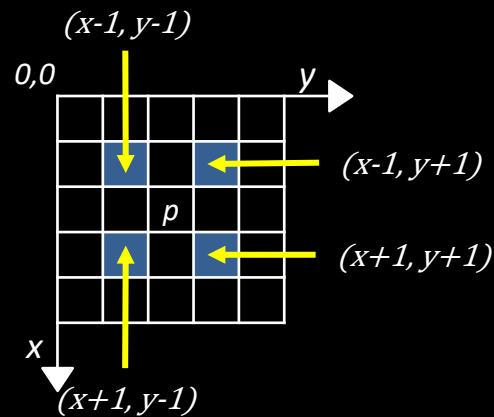
Vizinhança-4 de  $p$ ,  $N_4(p)$ :

$(x+1, y)$ ,  $(x-1, y)$ ,  $(x, y+1)$ ,  $(x, y-1)$



Vizinhança-diagonal de  $p$ ,  $N_D(p)$ :

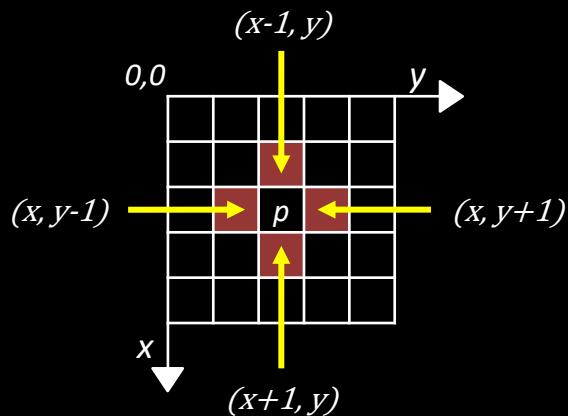
$(x-1, y-1)$ ,  $(x-1, y+1)$ ,  $(x+1, y-1)$ ,  
 $(x+1, y+1)$



# Vizinhança de um pixel

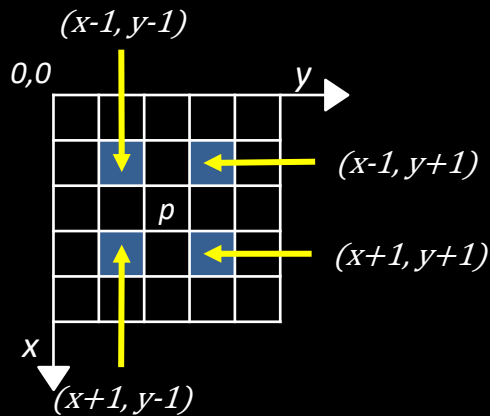
Vizinhança-4 de  $p$ ,  $N_4(p)$ :

$(x+1, y)$ ,  $(x-1, y)$ ,  $(x, y+1)$ ,  $(x, y-1)$

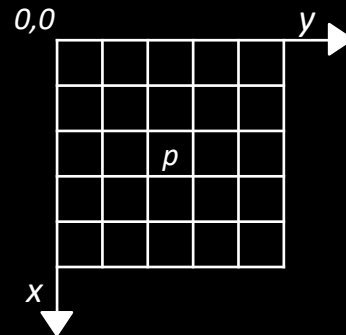


Vizinhança-diagonal de  $p$ ,  $N_D(p)$ :

$(x-1, y-1)$ ,  $(x-1, y+1)$ ,  $(x+1, y-1)$ ,  
 $(x+1, y+1)$



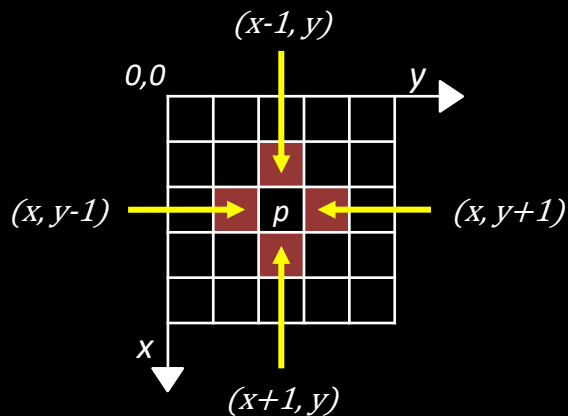
Vizinhança-8 de  $p$ ,  $N_8(p)$ :



# Vizinhança de um pixel

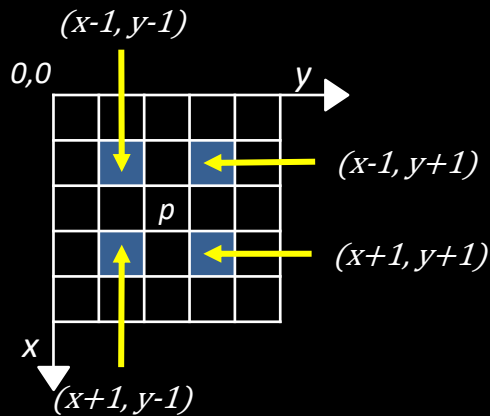
Vizinhança-4 de  $p$ ,  $N_4(p)$ :

$(x+1, y)$ ,  $(x-1, y)$ ,  $(x, y+1)$ ,  $(x, y-1)$



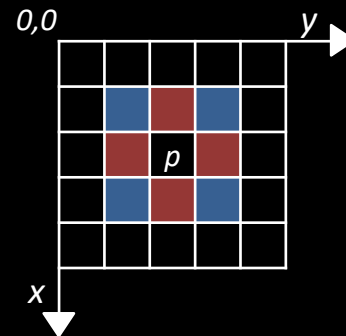
Vizinhança-diagonal de  $p$ ,  $N_D(p)$ :

$(x-1, y-1)$ ,  $(x-1, y+1)$ ,  $(x+1, y-1)$ ,  
 $(x+1, y+1)$



Vizinhança-8 de  $p$ ,  $N_8(p)$ :

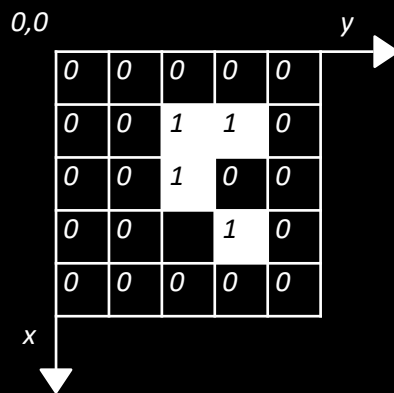
$N_4(p) \cup N_D(p)$



# Adjacência

## Adjacencia-4:

- Dois pixels  $p$  e  $q$  são adjacentes-4 se:
  - Os valores de  $p$  e  $q$  pertencem ao conjunto  $V$  e
  - O pixel  $q$  está no conjunto  $N_4(p)$

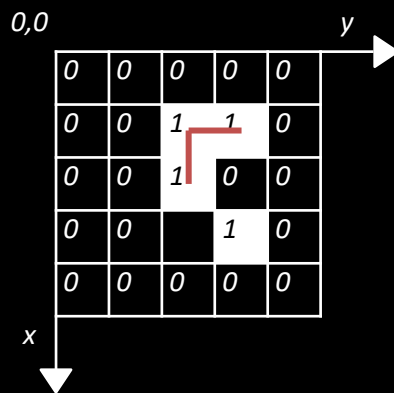


(\*)  $V = \{1\}$  para imagens binárias

# Adjacência

## Adjacencia-4:

- Dois pixels  $p$  e  $q$  são adjacentes-4 se:
  - Os valores de  $p$  e  $q$  pertencem ao conjunto  $V$  e
  - O pixel  $q$  está no conjunto  $N_4(p)$

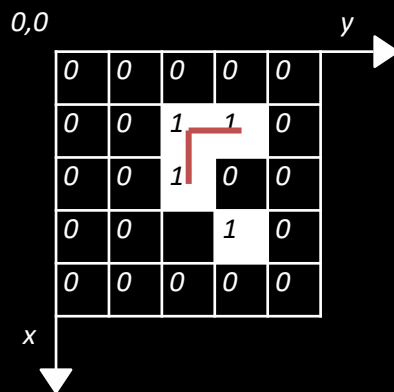


(\*)  $V = \{1\}$  para imagens binárias

# Adjacência

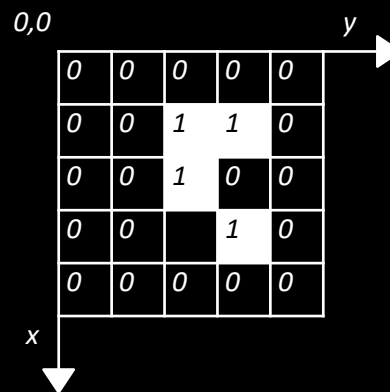
## Adjacencia-4:

- Dois pixels  $p$  e  $q$  são adjacentes-4 se:
  - Os valores de  $p$  e  $q$  pertencem ao conjunto  $V$  e
  - O pixel  $q$  está no conjunto  $N_4(p)$



## Adjacencia-8:

- Dois pixels  $p$  e  $q$  são adjacentes-8 se:
  - Os valores de  $p$  e  $q$  pertencem ao conjunto  $V$  e
  - O pixel  $q$  está no conjunto  $N_8(p)$ .

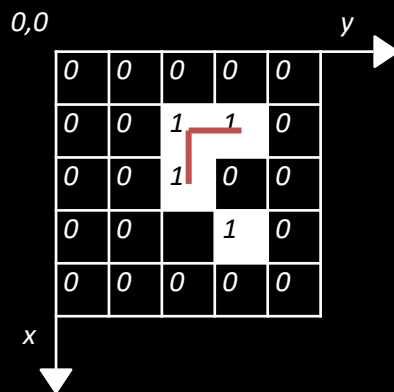


(\*)  $V = \{1\}$  para imagens binárias

# Adjacência

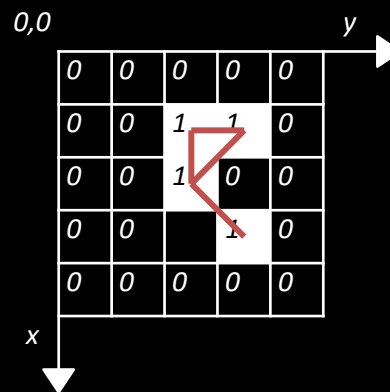
## Adjacência-4:

- Dois pixels  $p$  e  $q$  são adjacentes-4 se:
  - Os valores de  $p$  e  $q$  pertencem ao conjunto  $V$  e
  - O pixel  $q$  está no conjunto  $N_4(p)$



## Adjacência-8:

- Dois pixels  $p$  e  $q$  são adjacentes-8 se:
  - Os valores de  $p$  e  $q$  pertencem ao conjunto  $V$  e
  - O pixel  $q$  está no conjunto  $N_8(p)$ .

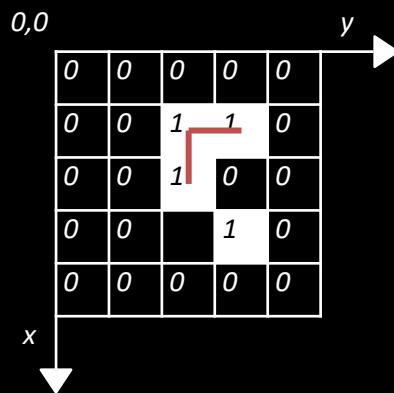


(\*)  $V = \{1\}$  para imagens binárias

# Adjacência

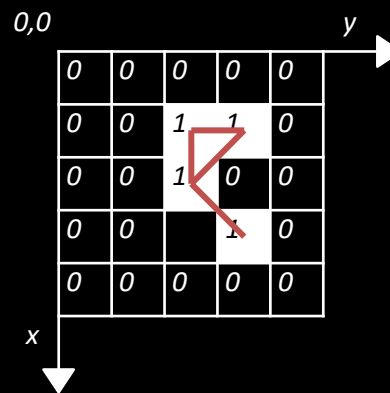
## Adjacência-4:

- Dois pixels  $p$  e  $q$  são adjacentes-4 se:
  - Os valores de  $p$  e  $q$  pertencem ao conjunto  $V$  e
  - O pixel  $q$  está no conjunto  $N_4(p)$



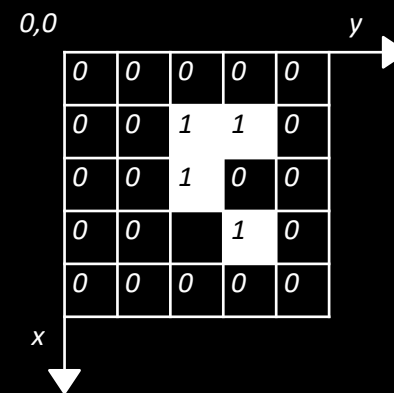
## Adjacência-8:

- Dois pixels  $p$  e  $q$  são adjacentes-8 se:
  - Os valores de  $p$  e  $q$  pertencem ao conjunto  $V$  e
  - O pixel  $q$  está no conjunto  $N_8(p)$ .



## Adjacência-m (adjacência mista):

- Dois pixels  $p$  e  $q$  são adjacentes-m se:
  - $q$  está em  $N_4(p)$  **OU**
  - $q$  estiver em  $N_D(p)$  e a intersecção entre  $N_4(p)$  e  $N_4(q)$  não contém nenhum pixel cujos valores pertencem a  $V$ .



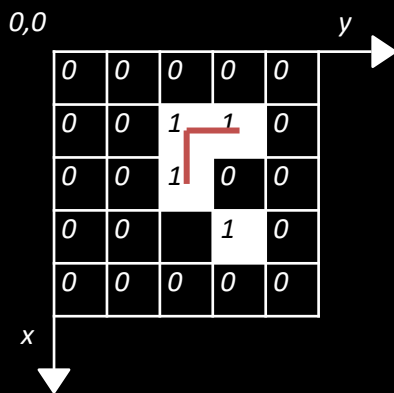
(\*)  $V = \{1\}$  para imagens binárias



# Adjacência

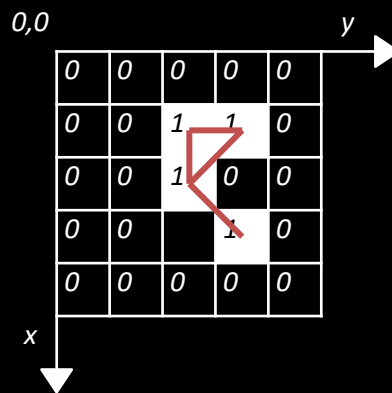
## Adjacência-4:

- Dois pixels  $p$  e  $q$  são adjacentes-4 se:
  - Os valores de  $p$  e  $q$  pertencem ao conjunto  $V$
  - O pixel  $q$  está no conjunto  $N_4(p)$



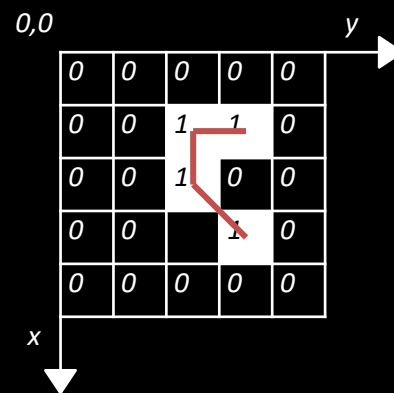
## Adjacência-8:

- Dois pixels  $p$  e  $q$  são adjacentes-8 se:
  - Os valores de  $p$  e  $q$  pertencem ao conjunto  $V$
  - O pixel  $q$  está no conjunto  $N_8(p)$ .



## Adjacência-m (adjacência mista):

- Dois pixels  $p$  e  $q$  são adjacentes-m se:
  - $q$  está em  $N_4(p)$  **OU**
  - $q$  estiver em  $N_D(p)$  e a intersecção entre  $N_4(p)$  e  $N_4(q)$  não contém nenhum pixel cujos valores pertencem a  $V$ .



(\*)  $V = \{1\}$  para imagens binárias

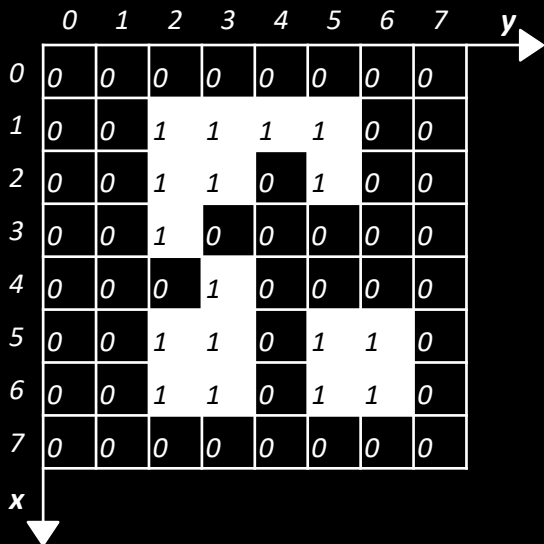
# Caminho (ou curva) digital

- Um **caminho** do pixel  $p$  com coordenadas  $(x, y)$  ao pixel  $q$  com coordenadas  $(s, t)$  é
  - uma sequencia de pixels distintos com coordenadas:
    - $(x_0, y_0), (x_1, y_1), \dots, (x_n, y_n)$
  - em que:
    - $(x_0, y_0) = (x, y)$ ,
    - $(x_n, y_n) = (s, t)$  e
    - os pixels  $(x_i, y_i)$  e  $(x_{i-1}, y_{i-1})$  são adjacentes para  $1 \leq i \leq n$
- Se  $(x_0, y_0) = (x_n, y_n)$ , o caminho é fechado
- Dependendo do tipo de adjacência escolhida, os caminhos podem ser:
  - caminho-4
  - caminho-8
  - caminho-m

# Caminho (ou curva) digital

Considerando vizinhança-4:

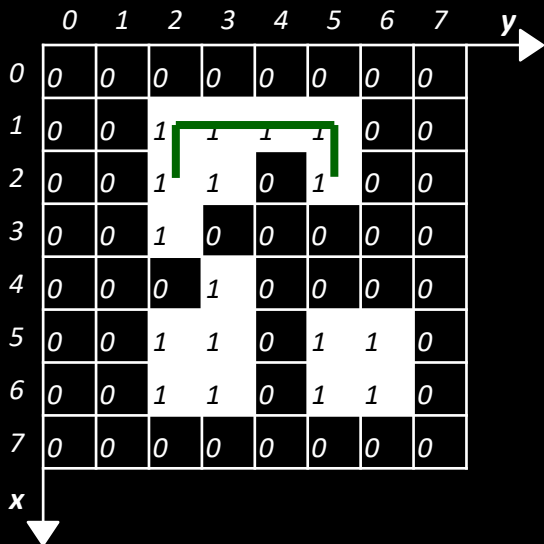
- Um dos caminhos entre  $p$  em  $(2,5)$  e  $q$  em  $(2,2)$ :



# Caminho (ou curva) digital

## Considerando vizinhança-4:

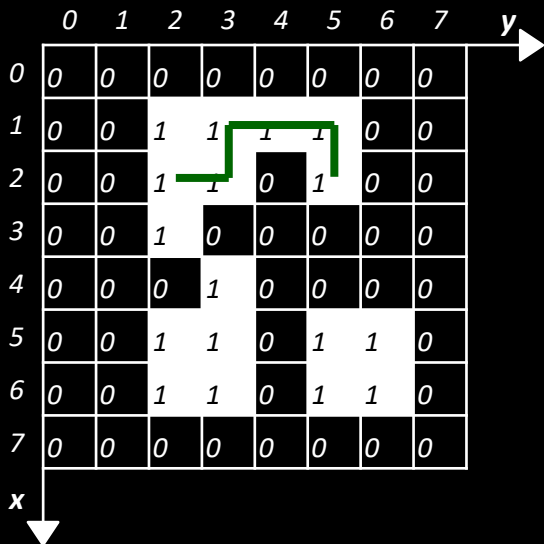
- Um dos caminhos entre  $p$  em  $(2,5)$  e  $q$  em  $(2,2)$ :
  - $(2,5), (1,5), (1,4), (1,3), (1,2), (2,2)$ .



# Caminho (ou curva) digital

## Considerando vizinhança-4:

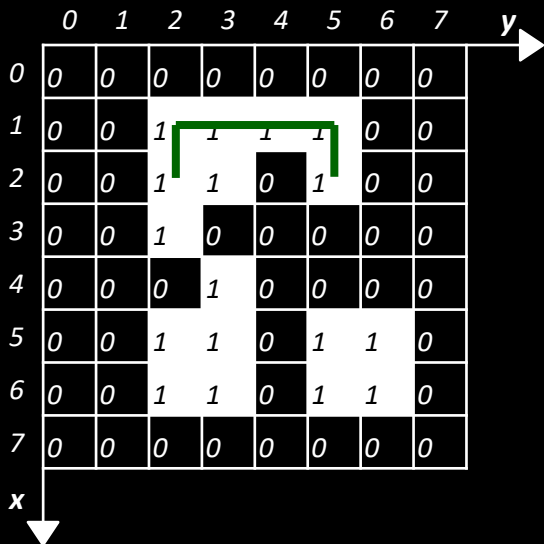
- Outros caminhos entre  $p$  em  $(2,5)$  e  $q$  em  $(2,2)$ :
  - $(2,5), (1,5), (1,4), (1,3), (2,3), (2,2)$ .



# Caminho (ou curva) digital

## Considerando vizinhança-4:

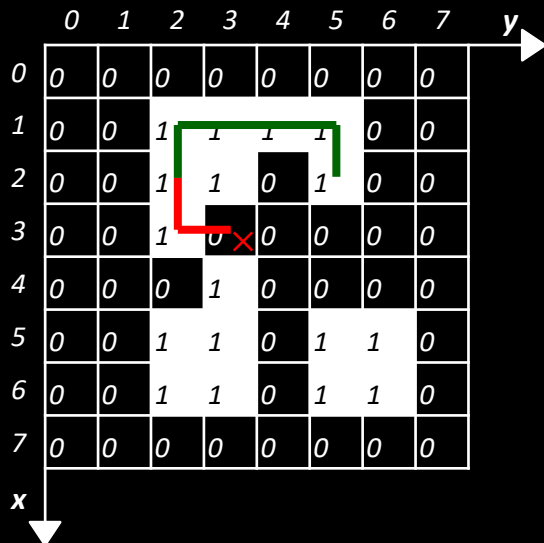
- Um dos caminhos entre  $p$  em  $(2,5)$  e  $q$  em  $(2,2)$ :
  - $(2,5), (1,5), (1,4), (1,3), (1,2), (2,2)$ .
- Um caminho entre  $p$  em  $(2,3)$  e  $q$  em  $(6,2)$ :



# Caminho (ou curva) digital

## Considerando vizinhança-4:

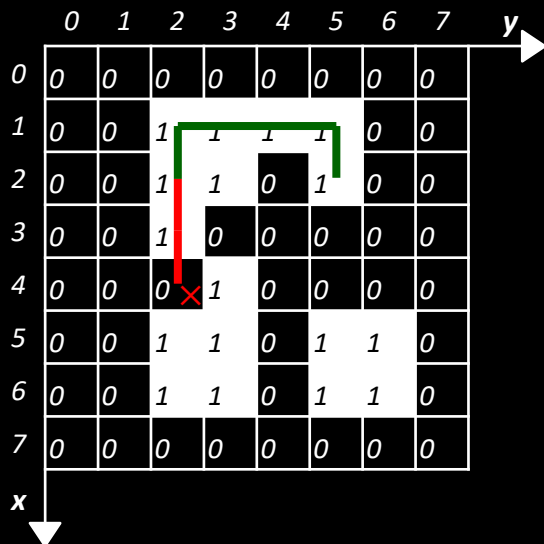
- Um dos caminhos entre  $p$  em  $(2,5)$  e  $q$  em  $(2,2)$ :
  - $(2,5), (1,5), (1,4), (1,3), (1,2), (2,2)$ .
- Um caminho entre  $p$  em  $(2,3)$  e  $q$  em  $(6,2)$ :



# Caminho (ou curva) digital

## Considerando vizinhança-4:

- Um dos caminhos entre  $p$  em  $(2,5)$  e  $q$  em  $(2,2)$ :
  - $(2,5), (1,5), (1,4), (1,3), (1,2), (2,2)$ .
- Um caminho entre  $p$  em  $(2,3)$  e  $q$  em  $(6,2)$ :

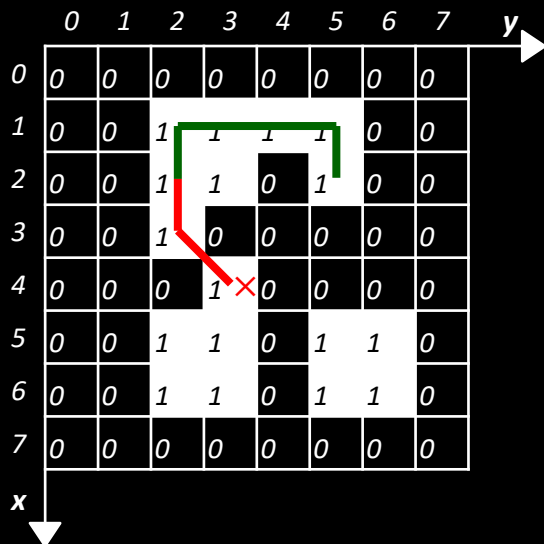




# Caminho (ou curva) digital

**Considerando vizinhança-4:**

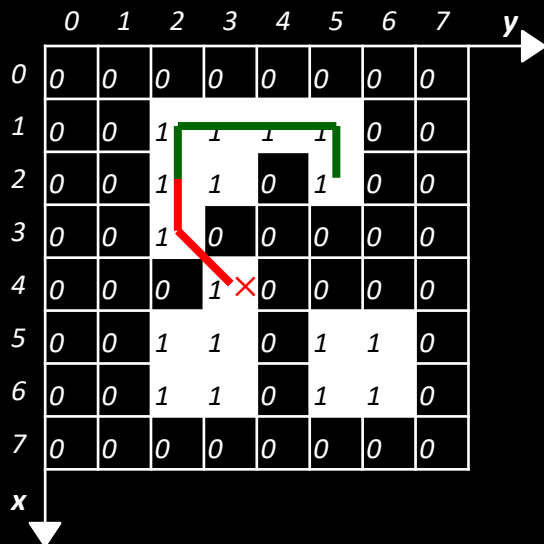
- Um dos caminhos entre  $p$  em  $(2,5)$  e  $q$  em  $(2,2)$ :
  - $(2,5), (1,5), (1,4), (1,3), (1,2), (2,2)$ .
- Um caminho entre  $p$  em  $(2,3)$  e  $q$  em  $(6,2)$ :
  - Não existe um caminho!



# Caminho (ou curva) digital

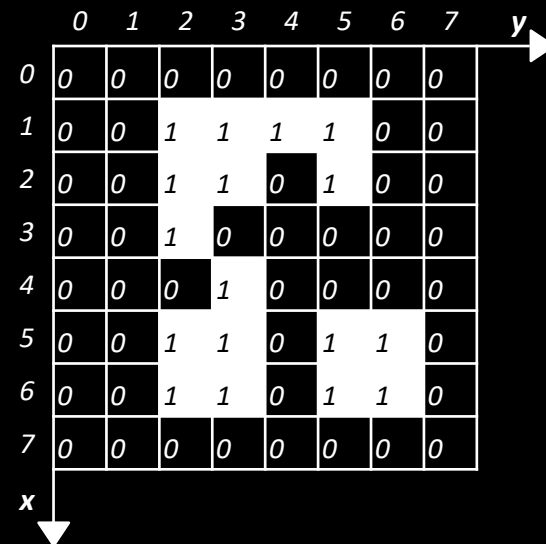
## Considerando vizinhança-4:

- Um dos caminhos entre  $p$  em  $(2,5)$  e  $q$  em  $(2,2)$ :
  - $(2,5), (1,5), (1,4), (1,3), (1,2), (2,2)$ .
- Um caminho entre  $p$  em  $(2,3)$  e  $q$  em  $(6,2)$ :
  - Não existe um caminho!



## Considerando vizinhança-8:

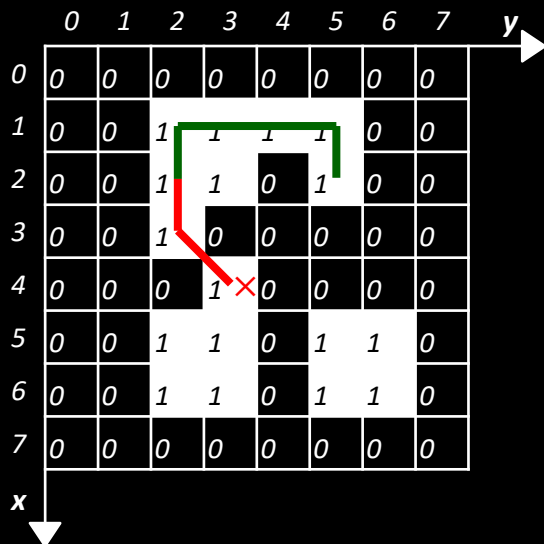
- Um dos caminhos entre  $p$  em  $(2,5)$  e  $q$  em  $(2,2)$ :



# Caminho (ou curva) digital

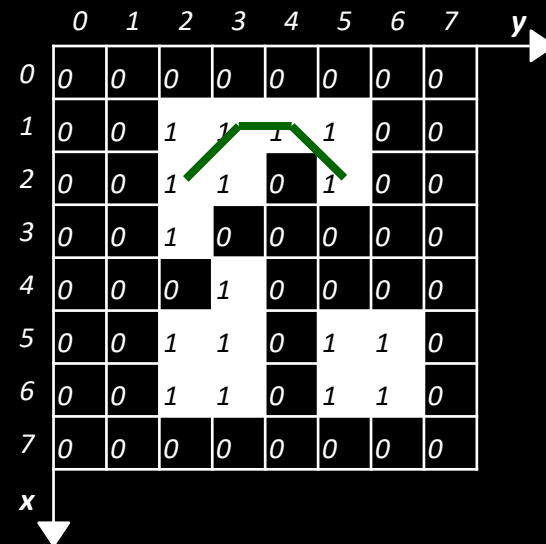
## Considerando vizinhança-4:

- Um dos caminhos entre  $p$  em (2,5) e  $q$  em (2,2):
  - (2,5), (1,5), (1,4), (1,3), (1,2), (2,2).
- Um caminho entre  $p$  em (2,3) e  $q$  em (6,2):
  - Não existe um caminho.



## Considerando vizinhança-8:

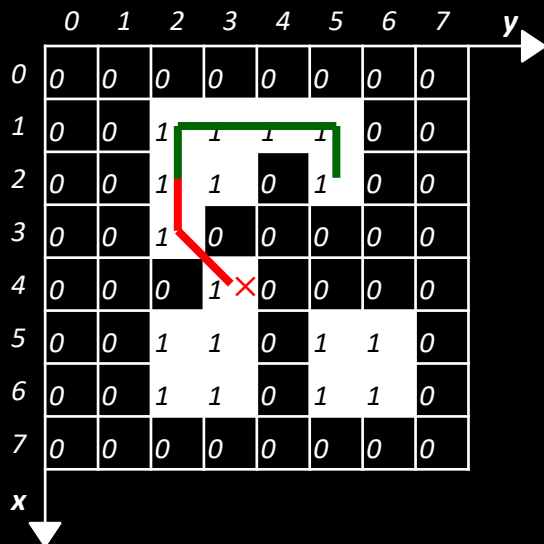
- Um dos caminhos entre  $p$  em (2,5) e  $q$  em (2,2):
  - (2,5), (1,4), (1,3), (2,2).



# Caminho (ou curva) digital

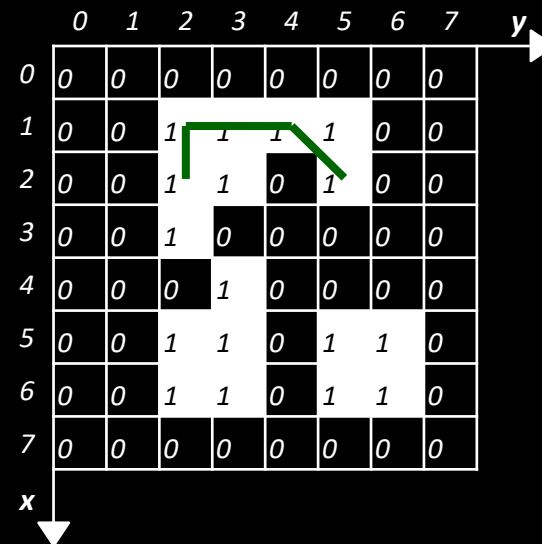
## Considerando vizinhança-4:

- Um dos caminhos entre  $p$  em (2,5) e  $q$  em (2,2):
  - (2,5), (1,5), (1,4), (1,3), (1,2), (2,2).
- Um caminho entre  $p$  em (2,3) e  $q$  em (6,2):
  - Não existe um caminho.



## Considerando vizinhança-8:

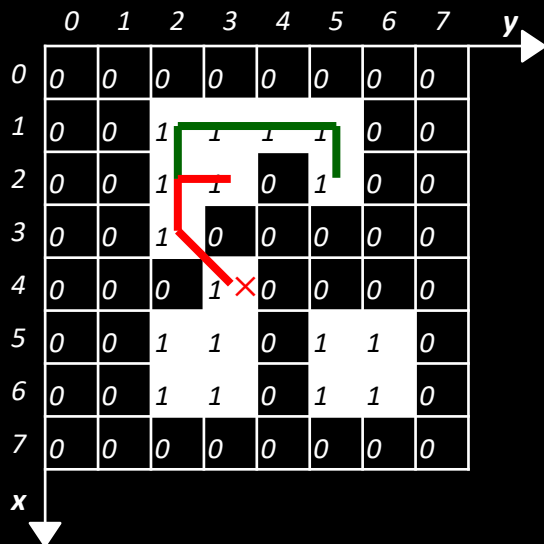
- Outro caminho entre  $p$  em (2,5) e  $q$  em (2,2):
  - (2,5), (1,4), (1,3), (1,2), (2,2).



# Caminho (ou curva) digital

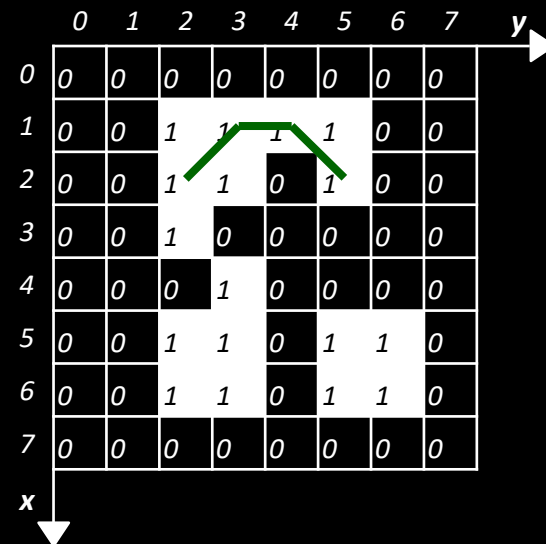
## Considerando vizinhança-4:

- Um dos caminhos entre  $p$  em (2,5) e  $q$  em (2,2):
  - (2,5), (1,5), (1,4), (1,3), (1,2), (2,2).
- Um caminho entre  $p$  em (2,3) e  $q$  em (6,2):
  - Não existe um caminho!



## Considerando vizinhança-8:

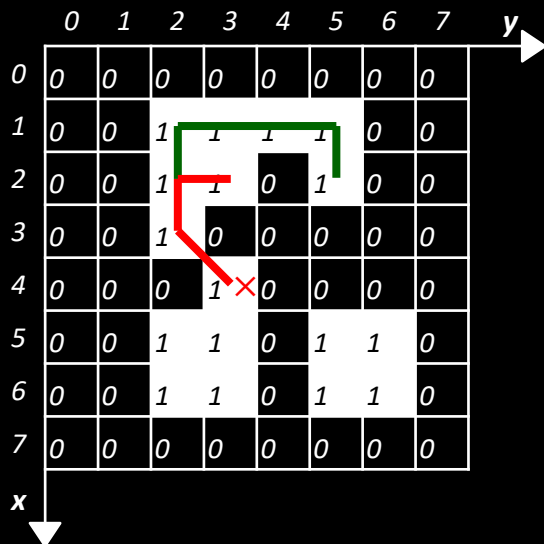
- Um dos caminhos entre  $p$  em (2,5) e  $q$  em (2,2):
  - (2,5), (1,4), (1,3), (2,2).
- Um dos caminhos entre  $p$  em (2,3) e  $q$  em (6,2):



# Caminho (ou curva) digital

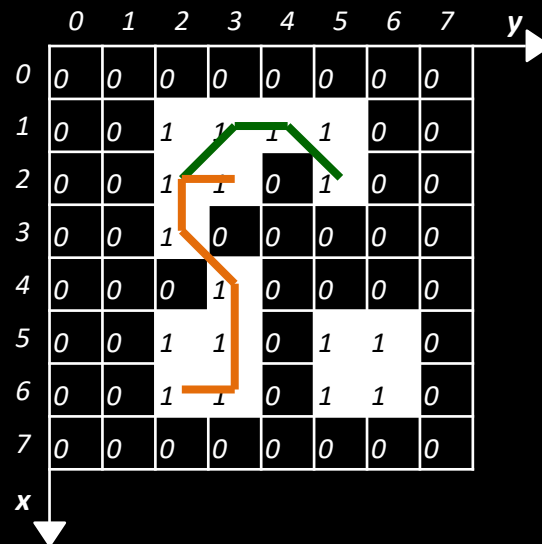
## Considerando vizinhança-4:

- Um dos caminhos entre  $p$  em  $(2,5)$  e  $q$  em  $(2,2)$ :
  - $(2,5), (1,5), (1,4), (1,3), (1,2), (2,2)$ .
- Um caminho entre  $p$  em  $(2,3)$  e  $q$  em  $(6,2)$ :
  - Não existe um caminho!



## Considerando vizinhança-8:

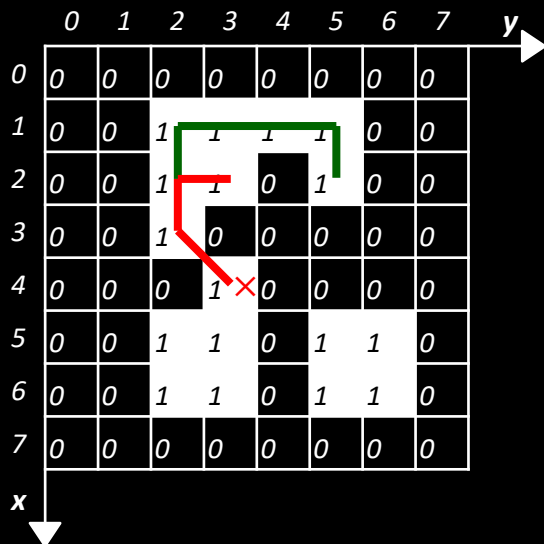
- Um dos caminhos entre  $p$  em  $(2,5)$  e  $q$  em  $(2,2)$ :
  - $(2,5), (1,4), (1,3), (2,2)$ .
- Um dos caminhos entre  $p$  em  $(2,3)$  and  $q$  em  $(6,2)$ :
  - $(2,3), (2,2), (3,2), (4,3), (5,3), (6,3), (6,2)$ .



# Caminho (ou curva) digital

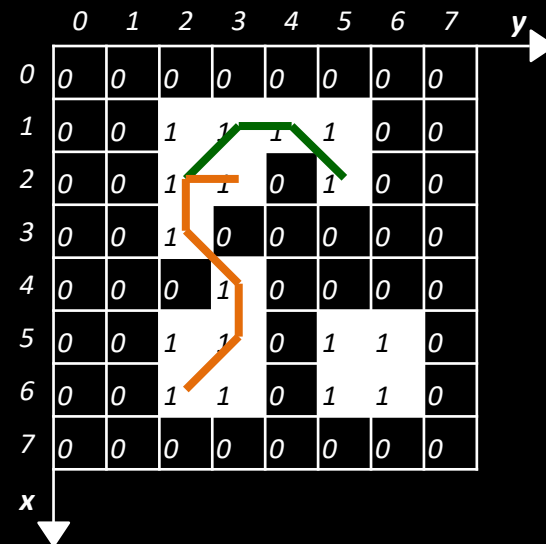
## Considerando vizinhança-4:

- Um dos caminhos entre  $p$  em (2,5) e  $q$  em (2,2):
  - (2,5), (1,5), (1,4), (1,3), (1,2), (2,2).
- Um caminho entre  $p$  em (2,3) e  $q$  em (6,2):
  - Não existe um caminho!



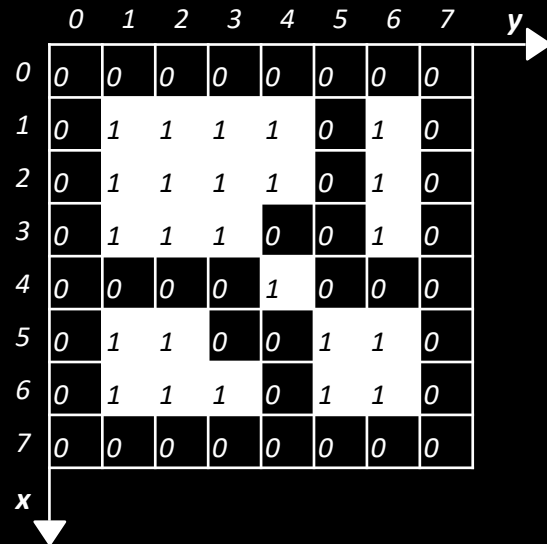
## Considerando vizinhança-8:

- Um dos caminhos entre  $p$  em (2,5) e  $q$  em (2,2):
  - (2,5), (1,4), (1,3), (2,2).
- Outro caminho entre  $p$  em (2,3) e  $q$  em (6,2):
  - (2,3), (2,2), (3,2), (4,3), (5,3), (6,2).



# Regiões conectadas e componentes conectados

- **Região conectada:**
  - Qualquer região  $R$  que existe pelo menos um caminho entre quaisquer pares de pixels  $(p, q)$
- **Componente conectado:**
  - Região conectada máxima
  - Não é um subconjunto próprio de nenhuma região conectada maior





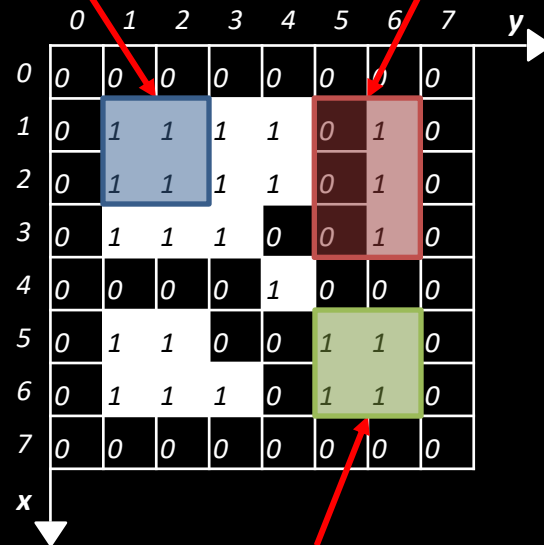
# Regiões conectadas e componentes conectados

- **Região conectada:**
  - Qualquer região  $R$  que existe pelo menos um caminho entre quaisquer pares de pixels  $(p, q)$

- **Componente conectado:**
  - Região conectada máxima
  - Não é um subconjunto próprio de nenhuma região conectada maior

Região não-conectada

Região conectada não máxima



Região conectada máxima  
(Componente conectado)

# Componentes conectados

Imagem binária

	0	1	2	3	4	5	6	7	y
0	0	0	0	0	0	0	0	0	
1	0	1	1	1	1	0	1	0	
2	0	1	1	1	1	0	1	0	
3	0	1	1	1	0	0	1	0	
4	0	0	0	0	1	0	0	0	
5	0	1	1	0	0	1	1	0	
6	0	1	1	1	0	1	1	0	
7	0	0	0	0	0	0	0	0	
x									

$N_4(p)$

	0	1	2	3	4	5	6	7	y
0	0	0	0	0	0	0	0	0	
1	0	1	1	1	1	0	1	0	
2	0	1	1	1	1	0	1	0	
3	0	1	1	1	0	0	1	0	
4	0	0	0	0	1	0	0	0	
5	0	1	1	0	0	1	1	0	
6	0	1	1	1	0	1	1	0	
7	0	0	0	0	0	0	0	0	
x									

$N_8(p)$

# Componentes conectados

Imagem binária

	0	1	2	3	4	5	6	7	y
0	0	0	0	0	0	0	0	0	
1	0	1	1	1	1	0	1	0	
2	0	1	1	1	1	0	1	0	
3	0	1	1	1	0	0	1	0	
4	0	0	0	0	1	0	0	0	
5	0	1	1	0	0	1	1	0	
6	0	1	1	1	0	1	1	0	
7	0	0	0	0	0	0	0	0	
x									

$N_4(p)$

	0	1	2	3	4	5	6	7	y
0	0	0	0	0	0	0	0	0	
1	0	1	1	1	1	0	1	0	
2	0	1	1	1	1	0	1	0	
3	0	1	1	1	0	0	1	0	
4	0	0	0	0	1	0	0	0	
5	0	1	1	0	0	1	1	0	
6	0	1	1	1	0	1	1	0	
7	0	0	0	0	0	0	0	0	
x									

$N_8(p)$

# Componentes conectados

Imagem binária

	0	1	2	3	4	5	6	7	y
0	0	0	0	0	0	0	0	0	
1	0	1	1	1	1	0	1	0	
2	0	1	1	1	1	0	1	0	
3	0	1	1	1	0	0	1	0	
4	0	0	0	0	1	0	0	0	
5	0	1	1	0	0	1	1	0	
6	0	1	1	1	0	1	1	0	
7	0	0	0	0	0	0	0	0	
x									

$N_4(p)$

	0	1	2	3	4	5	6	7	y
0	0	0	0	0	0	0	0	0	
1	0	1	1	1	1	0	1	0	
2	0	1	1	1	1	0	1	0	
3	0	1	1	1	0	0	1	0	
4	0	0	0	0	1	0	0	0	
5	0	1	1	0	0	1	1	0	
6	0	1	1	1	0	1	1	0	
7	0	0	0	0	0	0	0	0	
x									

$N_8(p)$

# Componentes conectados

Imagem binária

	0	1	2	3	4	5	6	7	y
0	0	0	0	0	0	0	0	0	
1	0	1	1	1	1	0	1	0	
2	0	1	1	1	1	0	1	0	
3	0	1	1	1	0	0	1	0	
4	0	0	0	0	1	0	0	0	
5	0	1	1	0	0	1	1	0	
6	0	1	1	1	0	1	1	0	
7	0	0	0	0	0	0	0	0	
x									

$N_4(p)$

	0	1	2	3	4	5	6	7	y
0	0	0	0	0	0	0	0	0	
1	0	1	1	1	1	0	1	0	
2	0	1	1	1	1	0	1	0	
3	0	1	1	1	0	0	1	0	
4	0	0	0	0	1	0	0	0	
5	0	1	1	0	0	1	1	0	
6	0	1	1	1	0	1	1	0	
7	0	0	0	0	0	0	0	0	
x									

$N_8(p)$

# Componentes conectados

Imagem binária

	0	1	2	3	4	5	6	7	y
0	0	0	0	0	0	0	0	0	
1	0	1	1	1	1	0	1	0	
2	0	1	1	1	1	0	1	0	
3	0	1	1	1	0	0	1	0	
4	0	0	0	0	1	0	0	0	
5	0	1	1	0	0	1	1	0	
6	0	1	1	1	0	1	1	0	
7	0	0	0	0	0	0	0	0	
x									

$N_4(p)$

	0	1	2	3	4	5	6	7	y
0	0	0	0	0	0	0	0	0	
1	0	1	1	1	1	0	1	0	
2	0	1	1	1	1	0	1	0	
3	0	1	1	1	0	0	1	0	
4	0	0	0	0	1	0	0	0	
5	0	1	1	0	0	1	1	0	
6	0	1	1	1	0	1	1	0	
7	0	0	0	0	0	0	0	0	
x									

$N_8(p)$

# Componentes conectados

$N_4(p)$

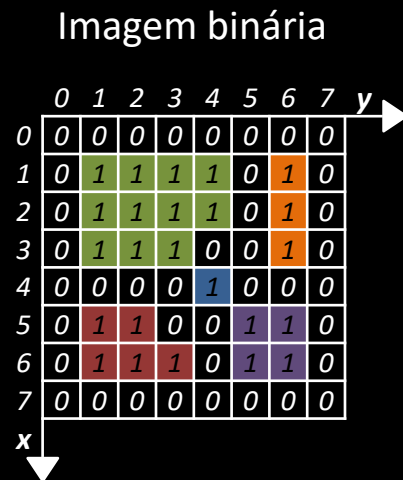
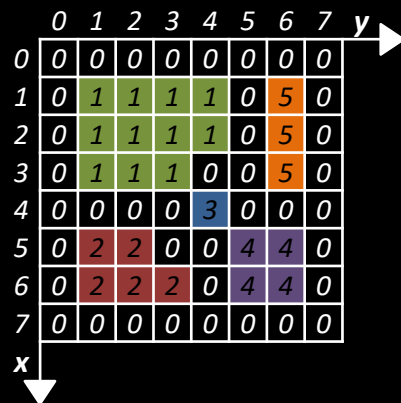
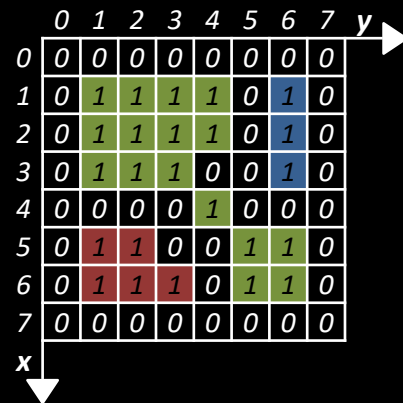


Imagem de rótulos



$N_8(p)$



# Componentes conectados

$N_4(p)$

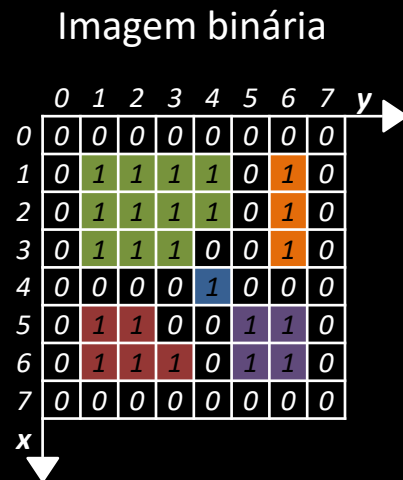
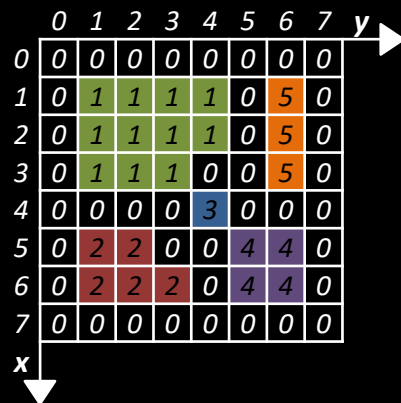
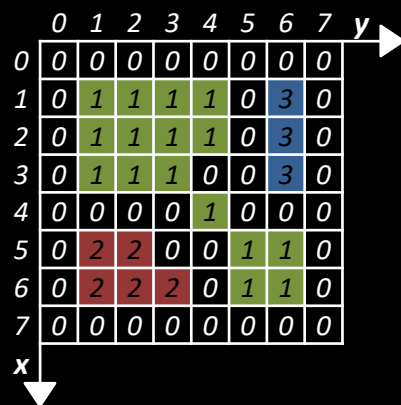
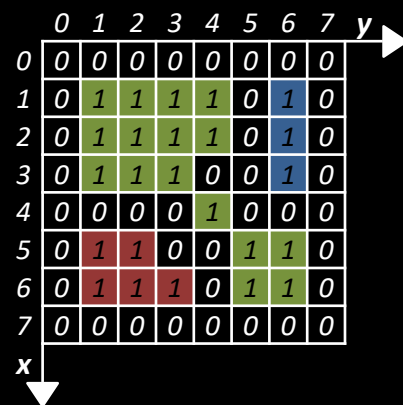


Imagem de rótulos



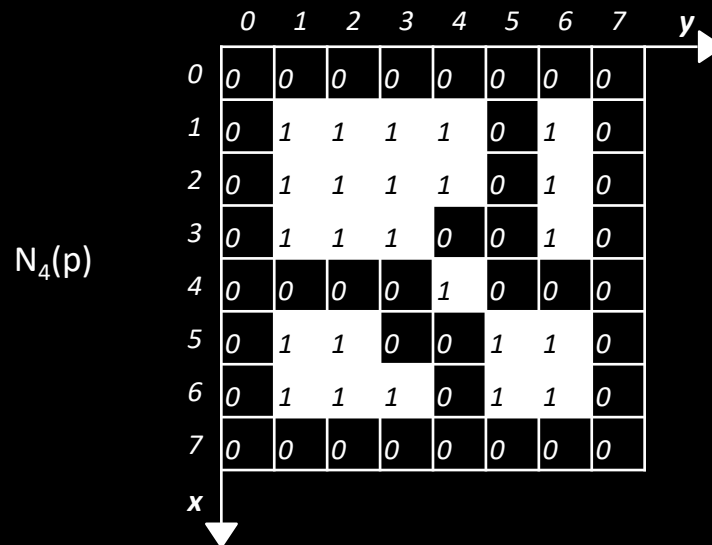
$N_8(p)$





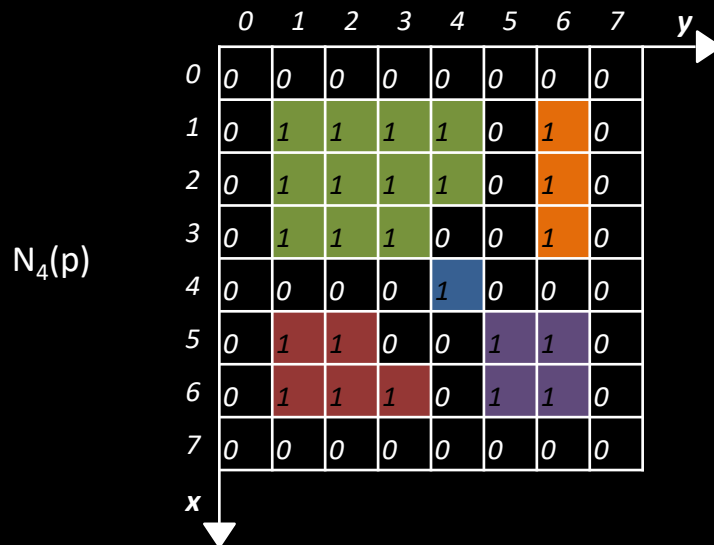
# Fundo e objetos de uma imagem

- Frente (*foreground*) da imagem (objetos)
  - Conjunto de todos os componentes conectados na imagem
- Fundo (*background*) da imagem
  - O complemento do conjunto dos componentes conectados



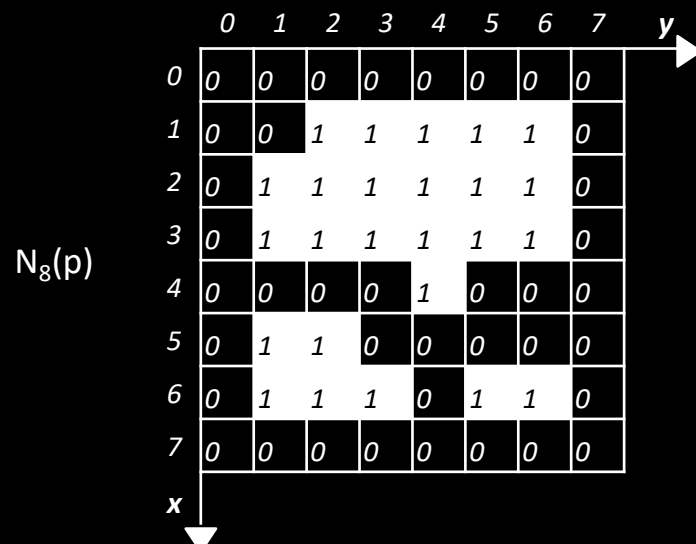
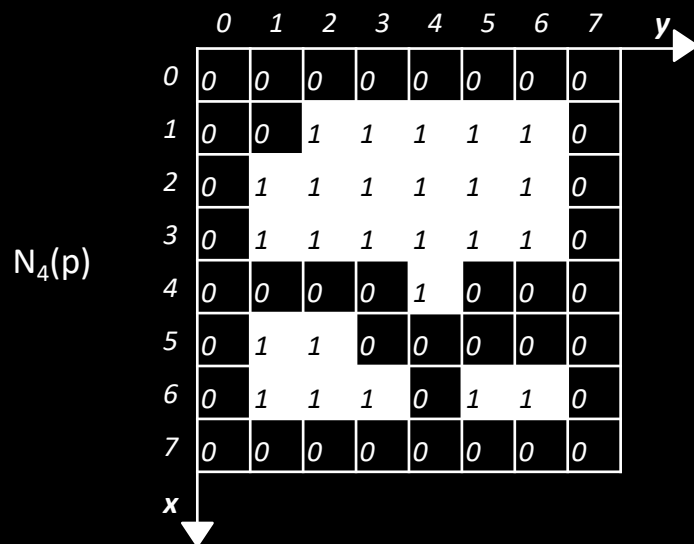
# Fundo e objetos de uma imagem

- Frente (*foreground*) da imagem (objetos)
  - Conjunto de todos os componentes conectados na imagem
- Fundo (*background*) da imagem
  - O complemento do conjunto dos componentes conectados



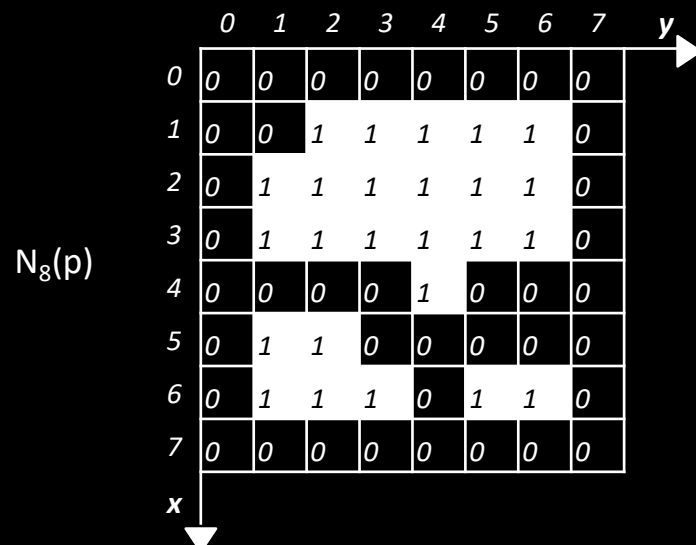
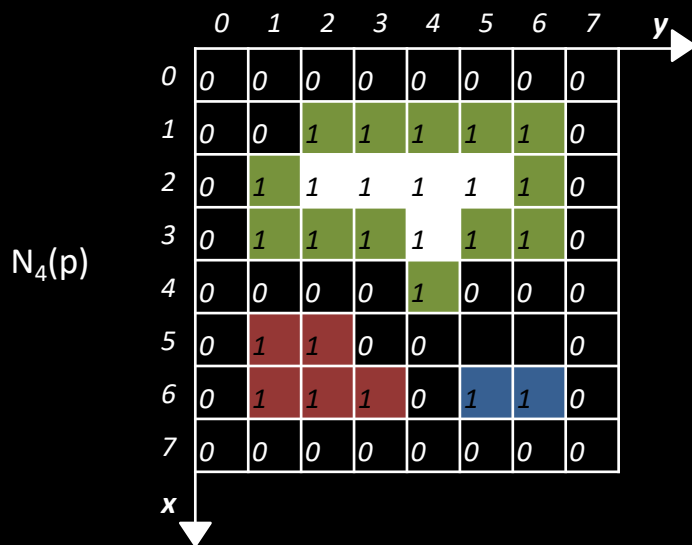
# Borda, contorno ou fronteira

- Borda de um componente conectado C:
  - Conjunto de pontos em C que são **adjacentes** aos pontos do complemento de C.
  - Dependente da conectividade.
  - Borda interna.



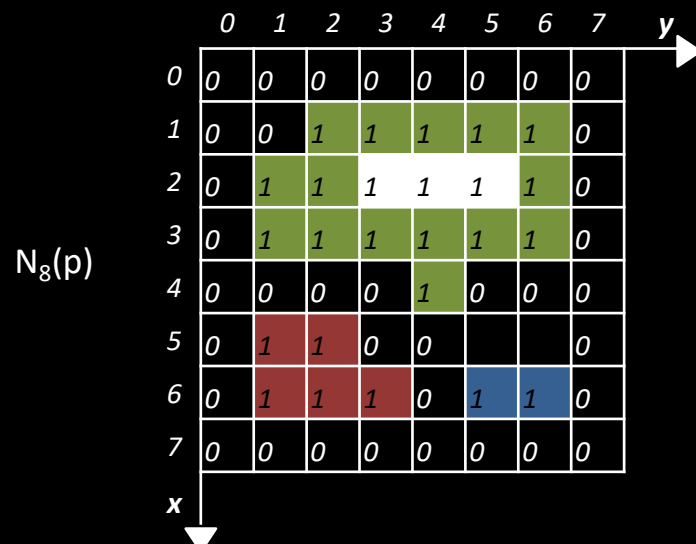
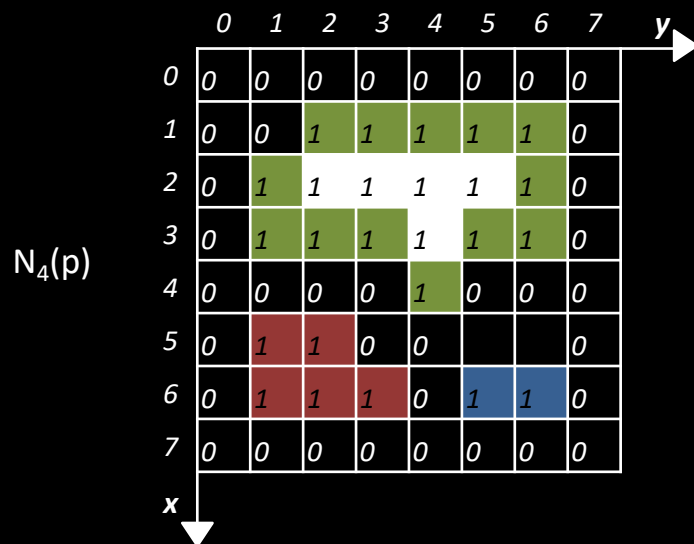
# Borda, contorno ou fronteira

- Borda de um componente conectado C:
  - Conjunto de pontos em C que são **adjacentes** aos pontos do complemento de C.
  - Dependente da conectividade.
  - Borda interna.



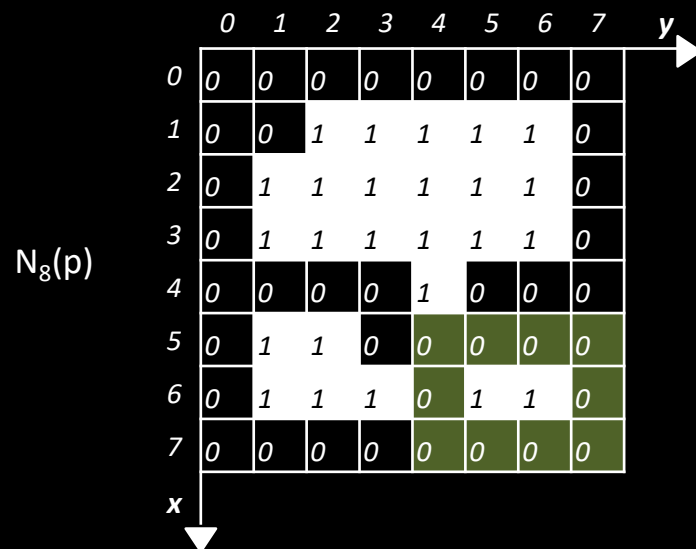
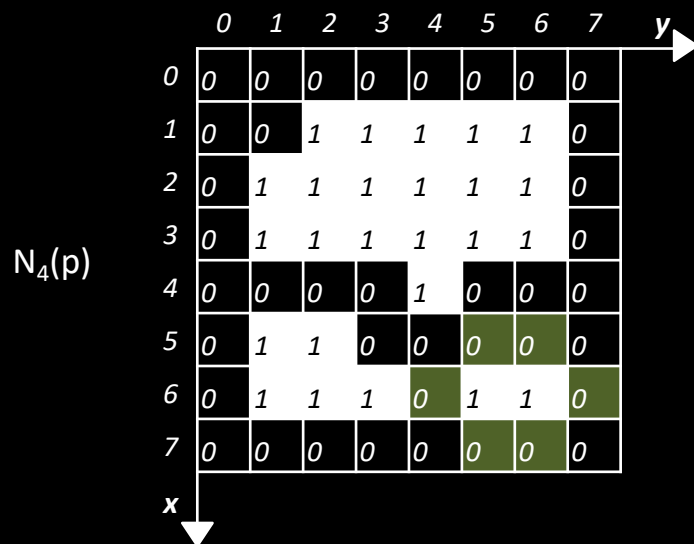
# Borda, contorno ou fronteira

- Borda de um componente conectado C:
  - Conjunto de pontos em C que são **adjacentes** aos pontos do complemento de C.
  - Dependente da conectividade.
  - Borda interna.



# Borda, contorno ou fronteira

- Borda **externas** de um componente conectado C:
  - Conjunto de pontos no complemento de C,  $C^c$ , que são **adjacentes** aos pontos em C.
  - Bordas sempre formam um conjunto fechado.
    - Algoritmos seguidores de contorno.



# OPERAÇÕES LÓGICAS E ARITMÉTICAS

# Operações aritméticas

- Operações aritméticas são realizadas entre pixels correspondentes
  - SOMA
    - $g(x, y) = f_1(x, y) + f_2(x, y)$
  - SUBTRAÇÃO
    - $g(x, y) = f_1(x, y) - f_2(x, y)$
  - MULTIPLICAÇÃO
    - $g(x, y) = f_1(x, y) \times f_2(x, y)$
  - DIVISÃO
    - $g(x, y) = f_1(x, y) / f_2(x, y)$



# Tipos de dados no Python (scikit-image)

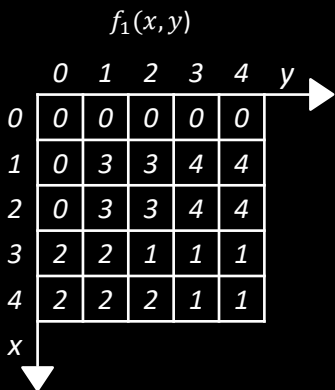
dtype	de	até	Descrição
uint8	0	255	Inteiro de 8 bits sem sinal
uint16	0	65,535	Inteiro de 16 bits sem sinal
uint32	0	4,294,967,295	Inteiro de 32 bits sem sinal
float	-1.0	+1.0	Ponto flutuante de 64 bits
int8	-128	127	Inteiro de 8 bits com sinal
int16	-32,768	+32,767	Inteiro de 16 bits com sinal
int32	$-2^{31}$	$2^{31} - 1$	Inteiro de 32 bits com sinal

Função	Descrição
img_as_float	Converte para float
img_as_ubyte	Converte para uint8
img_as_uint	Converte para uint16
img_as_int	Converte para int16

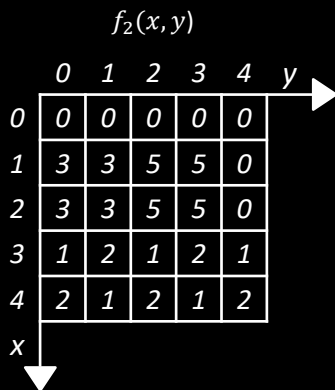
[https://scikit-image.org/docs/dev/user\\_guide/data\\_types.html](https://scikit-image.org/docs/dev/user_guide/data_types.html)

## SOMA

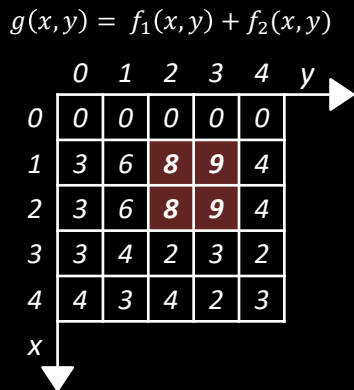
$k = 3$  (número de bits)  
 $L = 2^k = 2^3 = 8$   
Intervalo:  $[0, L-1]$  ou  $[0, 7]$



+

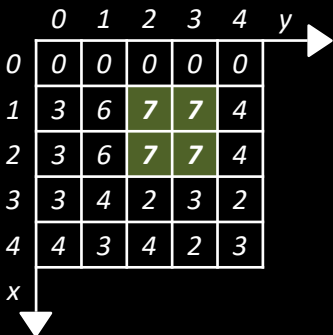


=



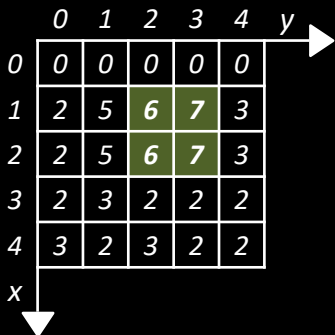
Truncamento:

$$g'(x, y) = \min(g(x, y), L - 1)$$



Normalização:

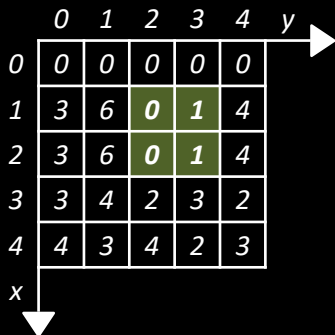
$$g' = \frac{L - 1}{g_{\max} - g_{\min}} \times (g - g_{\min})$$



<i>g</i>	<i>p</i> / 9 * 7	<i>p'</i>
0	0.00	0
1	0.77	1
2	1.55	2
3	2.33	2
4	3.11	3
5	3.88	4
6	4.66	5
7	5.44	5
8	6.22	6
9	7.00	7

Wrap-around:

$$g(x, y) > L - 1 ? g(x, y) - L : g(x, y)$$



# Operações aritméticas

## SUBTRAÇÃO

$k = 3$  (número de bits)

$L = 2^k = 2^3 = 8$

Intervalo:  $[0, L-1]$  ou  $[0, 7]$

$$f_1(x, y)$$

	0	1	2	3	4	y
0	0	0	0	0	0	
1	3	3	5	5	0	
2	3	3	5	5	0	
3	1	2	1	2	1	
4	2	1	2	1	2	
x						

—

$$f_2(x, y)$$

	0	1	2	3	4	y
0	0	0	0	0	0	
1	0	3	3	4	4	
2	0	3	3	4	4	
3	2	2	1	1	1	
4	2	2	2	1	1	
x						

=

$$g(x, y) = f_1(x, y) - f_2(x, y)$$

	0	1	2	3	4	y
0	0	0	0	0	0	
1	3	0	2	1	-4	
2	3	0	2	1	-4	
3	-1	0	0	1	0	
4	0	-1	0	0	1	
x						

Truncamento:

$$g'(x, y) = \max(g(x, y), 0)$$

	0	1	2	3	4	y
0	0	0	0	0	0	
1	3	0	2	1	0	
2	3	0	2	1	0	
3	0	0	0	1	0	
4	0	0	0	0	1	
x						

Normalização:

$$g' = \frac{L - 1}{g_{\max} - g_{\min}} \times (g - g_{\min})$$

	0	1	2	3	4	y
0	4	4	4	4	4	
1	7	4	6	5	0	
2	7	4	6	5	0	
3	3	4	4	5	4	
4	4	3	4	4	5	
x						

Valor absoluto:

$$g'(x, y) = |g(x, y)|$$

	0	1	2	3	4	y
0	0	0	0	0	0	
1	3	0	2	1	4	
2	3	0	2	1	4	
3	1	0	0	1	0	
4	0	1	0	0	1	
x						

Wrap-around:

$$g(x, y) < 0 ? L + g(x, y) : g(x, y)$$

	0	1	2	3	4	y
0	0	0	0	0	0	
1	3	0	2	1	4	
2	3	0	2	1	4	
3	7	0	0	1	0	
4	0	7	0	0	1	
x						

# Operações aritméticas

## MULTIPLICAÇÃO

$k = 3$  (número de bits)

$L = 2^k = 2^3 = 8$

Intervalo:  $[0, L-1]$  ou  $[0, 7]$

$$f_1(x, y)$$

	0	1	2	3	4	y
0	0	0	0	0	0	
1	3	3	5	5	0	
2	3	3	5	5	0	
3	1	2	1	2	1	
4	2	1	2	1	2	
x						

**X**

$$f_2(x, y)$$

	0	1	2	3	4	y
0	0	0	0	0	0	
1	0	3	3	4	4	
2	0	3	3	4	4	
3	2	2	1	1	1	
4	2	2	2	1	1	
x						

**=**

$$g(x, y) = f_1(x, y) \times f_2(x, y)$$

	0	1	2	3	4	y
0	0	0	0	0	0	
1	0	9	15	25	0	
2	0	9	15	25	0	
3	2	4	1	4	1	
4	4	2	4	2	2	
x						

(\*) Corrigir como fizemos com a SOMA e SUBTRAÇÃO

## MULTIPLICAÇÃO

*Mascaramento*

$$f_1(x, y)$$

	0	1	2	3	4	y
0	0	0	0	0	0	
1	3	3	5	5	0	
2	3	3	5	5	0	
3	1	2	1	2	1	
4	2	1	2	1	2	
x						

**X**

$$f_2(x, y)$$

	0	1	2	3	4	y
0	0	0	0	0	0	
1	0	1	1	0	0	
2	0	1	1	1	0	
3	0	1	1	1	0	
4	0	0	0	0	0	
x						

**=**

$$g(x, y) = f_1(x, y) \times f_2(x, y)$$

	0	1	2	3	4	y
0	0	0	0	0	0	
1	0	3	5	0	0	
2	0	3	5	5	0	
3	0	2	1	2	0	
4	0	0	0	0	0	
x						

# Operações aritméticas

## DIVISÃO

$k = 3$  (número de bits)

$L = 2^k = 2^3 = 8$

Intervalo:  $[0, L-1]$  ou  $[0, 7]$

$$f_1(x, y)$$

	0	1	2	3	4	y
0	0	0	0	0	0	
1	3	3	5	5	0	
2	3	3	5	5	0	
3	1	2	1	2	1	
4	2	1	2	1	2	
x						

/

$$f_2(x, y)$$

	0	1	2	3	4	y
0	0	0	0	0	0	
1	0	3	3	4	4	
2	0	3	3	4	4	
3	2	2	1	1	1	
4	2	2	2	1	1	
x						

=

$$g(x, y) = f_1(x, y) / f_2(x, y)$$

	0	1	2	3	4	y
0	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	
1	$\infty$	1.	1.66	1.25	0	
2	$\infty$	1.	1.66	1.25	0	
3	0.5	1.	1.	1.	1.	
4	1.	0.5	1.	1.	2.	
x						

# Operações aritméticas

## DIVISÃO

$k = 3$  (número de bits)

$L = 2^k = 2^3 = 8$

Intervalo:  $[0, L-1]$  ou  $[0, 7]$

$$f_1(x, y)$$

	0	1	2	3	4	y
0	0	0	0	0	0	
1	3	3	5	5	0	
2	3	3	5	5	0	
3	1	2	1	2	1	
4	2	1	2	1	2	
x						

/

$$f_2(x, y)$$

	0	1	2	3	4	y
0	0	0	0	0	0	
1	0	3	3	4	4	
2	0	3	3	4	4	
3	2	2	1	1	1	
4	2	2	2	1	1	
x						

=

$$g(x, y) = f_1(x, y) / f_2(x, y)$$

	0	1	2	3	4	y
0	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	
1	$\infty$	1.	1.66	1.25	0	
2	$\infty$	1.	1.66	1.25	0	
3	0.5	1.	1.	1.	1.	
4	1.	0.5	1.	1.	2.	
x						

## Divisão por zero

Converter para float

Substituir o 0 (zero) pelo menor valor positivo.

$\epsilon = \text{np.spacing}(1)$

$$f_2(x, y)'$$

	0	1	2	3	4	y
0	$\epsilon$	$\epsilon$	$\epsilon$	$\epsilon$	$\epsilon$	
1	$\epsilon$	3.	3.	4.	4.	
2	$\epsilon$	3.	3.	4.	4.	
3	2.	2.	1.	1.	1.	
4	2.	2.	2.	1.	1.	
x						

# Operações aritméticas

## DIVISÃO

$k = 3$  (número de bits)

$L = 2^k = 2^3 = 8$

Intervalo:  $[0, L-1]$  ou  $[0, 7]$

## Divisão por zero

Converter para float

Substituir o 0 (zero) por um valor positivo muito pequeno.

$\epsilon = \text{np.spacing}(1)$

$$f_1(x, y)$$

	0	1	2	3	4	y
0	0	0	0	0	0	
1	3	3	5	5	0	
2	3	3	5	5	0	
3	1	2	1	2	1	
4	2	1	2	1	2	
x						

/

$$f_2(x, y)$$

	0	1	2	3	4	y
0	0	0	0	0	0	
1	0	3	3	4	4	
2	0	3	3	4	4	
3	2	2	1	1	1	
4	2	2	2	1	1	
x						

=

$$g(x, y) = f_1(x, y) / f_2(x, y)$$

	0	1	2	3	4	y
0	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	
1	$\infty$	1.	1.66	1.25	0	
2	$\infty$	1.	1.66	1.25	0	
3	0.5	1.	1.	1.	1.	
4	1.	0.5	1.	1.	2.	
x						

$f_2(x, y)'$

	0	1	2	3	4	y
0	$\epsilon$	$\epsilon$	$\epsilon$	$\epsilon$	$\epsilon$	
1	$\epsilon$	3.	3.	4.	4.	
2	$\epsilon$	3.	3.	4.	4.	
3	2.	2.	1.	1.	1.	
4	2.	2.	2.	1.	1.	
x						

$g(x, y)' = f_1(x, y) / f_2(x, y)$

	0	1	2	3	4	y
0	0	0	0	0	0	
1	0	1	2	1	0	
2	0	1	2	1	0	
3	1	1	1	1	1	
4	1	1	1	1	2	
x						

Converter o resultado para inteiro (arredondar ou truncar).  
Tratar valores.

# Operações lógicas

- Operações lógicas ocorrem entre imagens binárias
  - Pixels == 0  $\rightarrow$  False
  - Pixel == 1  $\rightarrow$  True

A	B	NOT A	A AND B	A OR B	A NAND B	A NOR B	A XOR B
0	0	1	0	0	1	0	0
0	1	1	0	1	1	0	1
1	0	0	0	1	1	0	1
1	1	0	1	1	0	1	0



# Operações lógicas

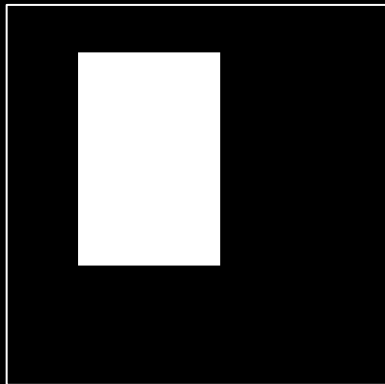


Imagem A

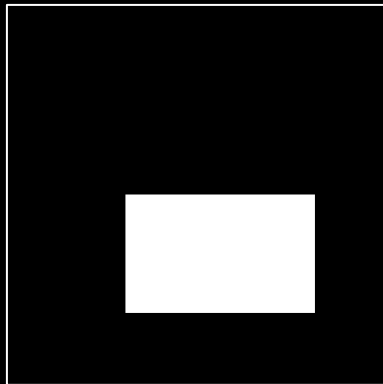
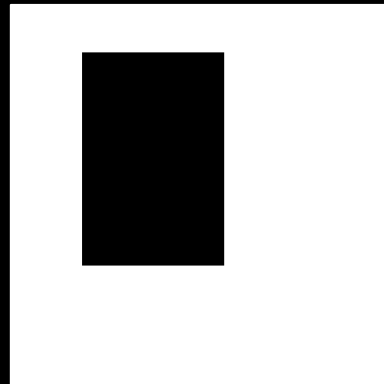
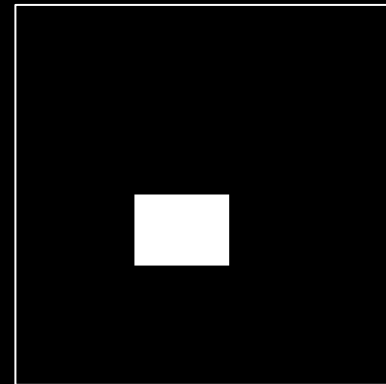


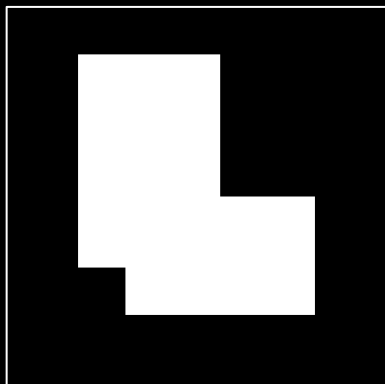
Imagem B



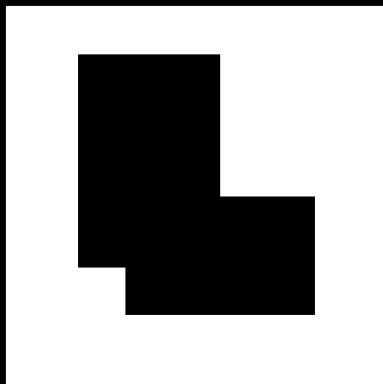
NOT A



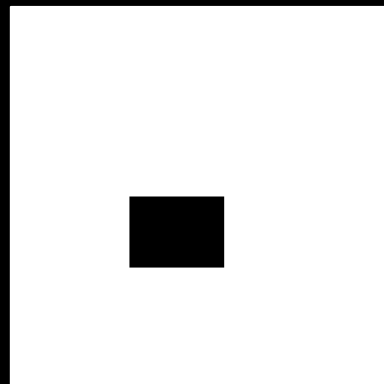
A AND B



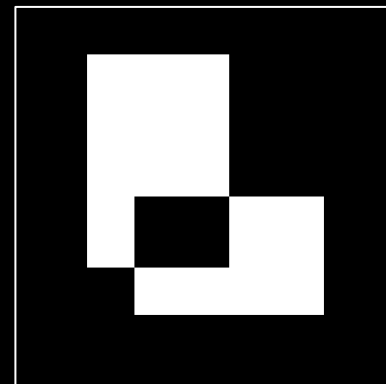
A OR B



A NOR B



A NAND B

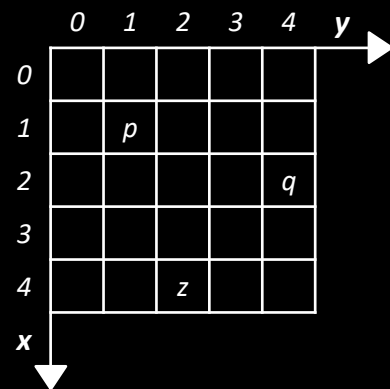


A XOR B

# MEDIDAS DE DISTÂNCIA

# Medidas de distância

- Considere três pixels e suas respectivas coordenadas
  - $p$  em  $(x, y)$ ,  $q$  em  $(s, t)$  e  $z$  em  $(v, w)$
- $D$  é uma função ou medida de distância
  - $D(p, q) \geq 0$ 
    - $D(p, q) = 0$  se  $p = q$
  - $D(p, q) = D(q, p)$
  - $D(p, z) \leq D(p, q) + D(q, z)$
- Algumas medidas de distância:
  - Distância Euclidiana
  - Distância *city block*
  - Distância *chessboard*



# Medidas de distância

- A distância Euclidiana entre os pixels  $p$  em  $(x, y)$  e  $q$  em  $(s, t)$

$$D_e(p, q) = \sqrt{(x - s)^2 + (y - t)^2}$$

	0	1	2	3	4	$y$
0	$\sqrt{8}$	$\sqrt{5}$	2	$\sqrt{5}$	$\sqrt{8}$	
1	$\sqrt{5}$	$\sqrt{2}$	1	$\sqrt{2}$	$\sqrt{5}$	
2	2	1	0	1	2	
3	$\sqrt{5}$	$\sqrt{2}$	1	$\sqrt{2}$	$\sqrt{5}$	
4	$\sqrt{8}$	$\sqrt{5}$	2	$\sqrt{5}$	$\sqrt{8}$	
$x$						

# Medidas de distância

- A distância Euclidiana entre os pixels  $p$  em  $(x, y)$  e  $q$  em  $(s, t)$

- $D_e(p, q) = \sqrt{(x - s)^2 + (y - t)^2}$

- Para  $p$  com coordenadas  $(2, 2)$  e:

- $q_1$  com coordenadas  $(1, 2)$ :

- $D_e(p, q) = \sqrt{(2 - 1)^2 + (2 - 2)^2}$

- $D_e(p, q) = \sqrt{1^2 + 0^2}$

- $D_e(p, q) = \sqrt{1} = 1$

- $q_2$  com coordenadas  $(1, 1)$ :

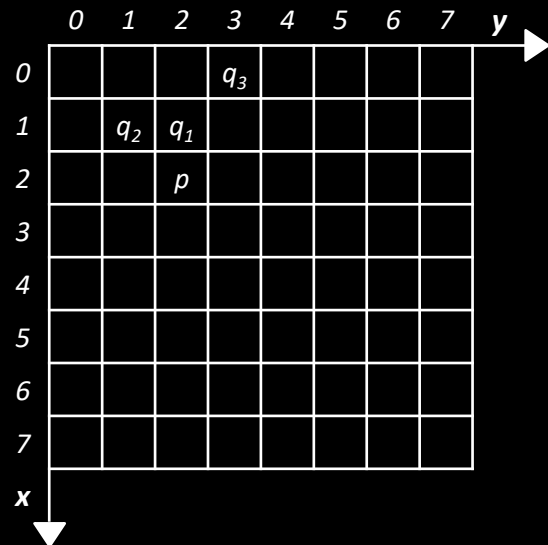
- $D_e(p, q) = \sqrt{(2 - 1)^2 + (2 - 1)^2}$

- $D_e(p, q) = \sqrt{1^2 + 1^2} = \sqrt{2}$

- $q_3$  com coordenadas  $(0, 3)$ :

- $D_e(p, q) = \sqrt{(2 - 0)^2 + (2 - 3)^2}$

- $D_e(p, q) = \sqrt{2^2 + (-1)^2} = \sqrt{5}$



# Medidas de distância

- A distância Euclidiana entre os pixels  $p$  em  $(x, y)$  e  $q$  em  $(s, t)$

- $D_e(p, q) = \sqrt{(x - s)^2 + (y - t)^2}$

- Para  $p$  com coordenadas  $(2, 2)$  e:

- $q_1$  com coordenadas  $(1, 2)$ :

- $D_e(p, q) = \sqrt{(2 - 1)^2 + (2 - 2)^2}$

- $D_e(p, q) = \sqrt{1^2 + 0^2}$

- $D_e(p, q) = \sqrt{1} = 1$

- $q_2$  com coordenadas  $(1, 1)$ :

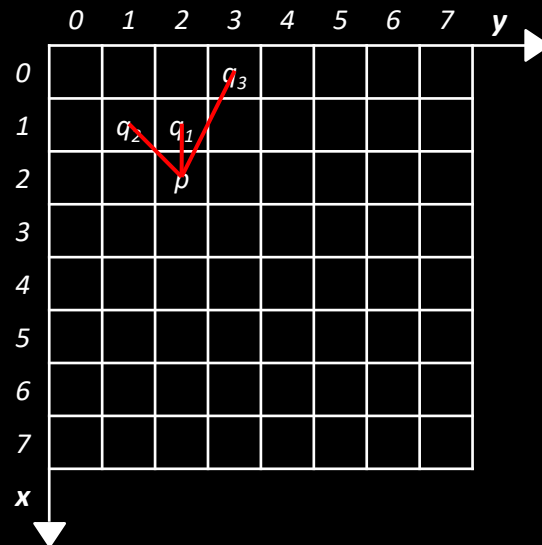
- $D_e(p, q) = \sqrt{(2 - 1)^2 + (2 - 1)^2}$

- $D_e(p, q) = \sqrt{1^2 + 1^2} = \sqrt{2}$

- $q_3$  com coordenadas  $(0, 3)$ :

- $D_e(p, q) = \sqrt{(2 - 0)^2 + (2 - 3)^2}$

- $D_e(p, q) = \sqrt{2^2 + (-1)^2} = \sqrt{5}$



# Medidas de distância

- A distância Euclidiana entre os pixels  $p$  em  $(x, y)$  e  $q$  em  $(s, t)$

- $D_e(p, q) = \sqrt{(x - s)^2 + (y - t)^2}$

- Para  $p$  com coordenadas  $(4, 3)$  e:

- $q_1$  com coordenadas  $(2, 2)$ :

- $D_e(p, q) = \sqrt{(4 - 2)^2 + (3 - 2)^2}$

- $D_e(p, q) = \sqrt{2^2 + 1^2} = \sqrt{5}$

- $q_2$  com coordenadas  $(5, 6)$ :

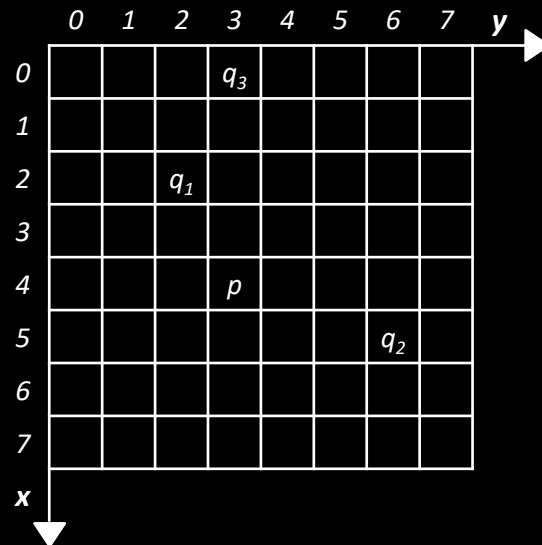
- $D_e(p, q) = \sqrt{(4 - 5)^2 + (3 - 6)^2}$

- $D_e(p, q) = \sqrt{(-1)^2 + (-3)^2} = \sqrt{1 + 9} = \sqrt{10}$

- $q_3$  com coordenadas  $(0, 3)$ :

- $D_e(p, q) = \sqrt{(4 - 0)^2 + (3 - 3)^2}$

- $D_e(p, q) = \sqrt{4^2 + 0^2} = \sqrt{16} = 4$



# Medidas de distância

- A distância Euclidiana entre os pixels  $p$  em  $(x, y)$  e  $q$  em  $(s, t)$

- $D_e(p, q) = \sqrt{(x - s)^2 + (y - t)^2}$

- Para  $p$  com coordenadas  $(4, 3)$  e:

- $q_1$  com coordenadas  $(2, 2)$ :

- $D_e(p, q) = \sqrt{(4 - 2)^2 + (3 - 2)^2}$

- $D_e(p, q) = \sqrt{2^2 + 1^2} = \sqrt{5}$

- $q_2$  com coordenadas  $(5, 6)$ :

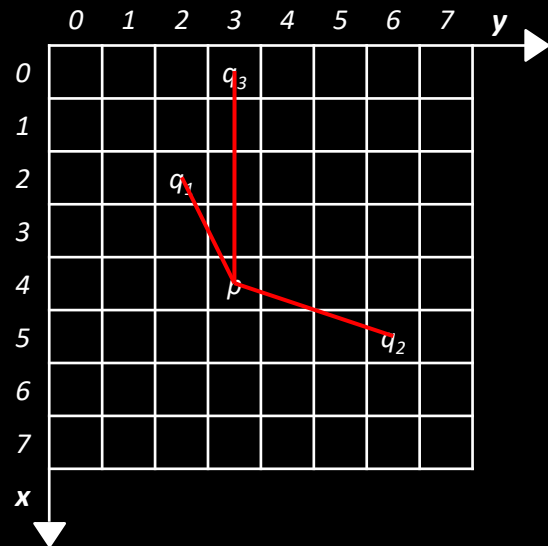
- $D_e(p, q) = \sqrt{(4 - 5)^2 + (3 - 6)^2}$

- $D_e(p, q) = \sqrt{(-1)^2 + (-3)^2} = \sqrt{1 + 9} = \sqrt{10}$

- $q_3$  com coordenadas  $(0, 3)$ :

- $D_e(p, q) = \sqrt{(4 - 0)^2 + (3 - 3)^2}$

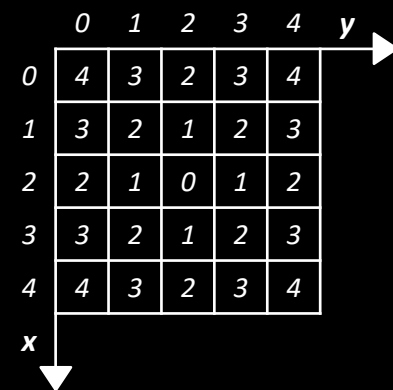
- $D_e(p, q) = \sqrt{4^2 + 0^2} = \sqrt{16} = 4$





# Medidas de distância

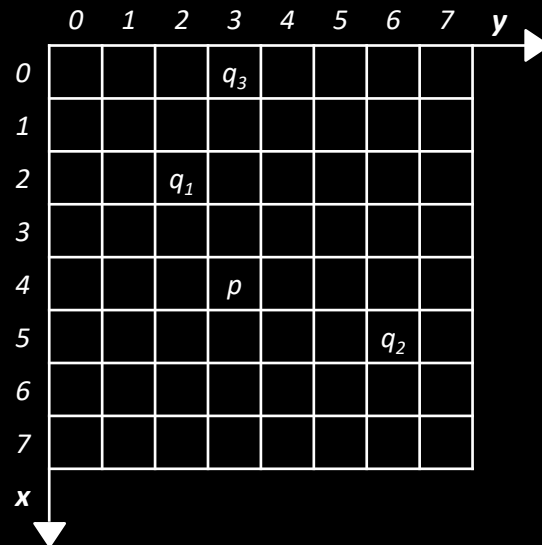
- Distância *city block* entre  $p$  em  $(x, y)$  e  $q$  em  $(s, t)$ 
  - $D_4(p, q) = |x - s| + |y - t|$



	0	1	2	3	4	y
0	4	3	2	3	4	
1	3	2	1	2	3	
2	2	1	0	1	2	
3	3	2	1	2	3	
4	4	3	2	3	4	
x						

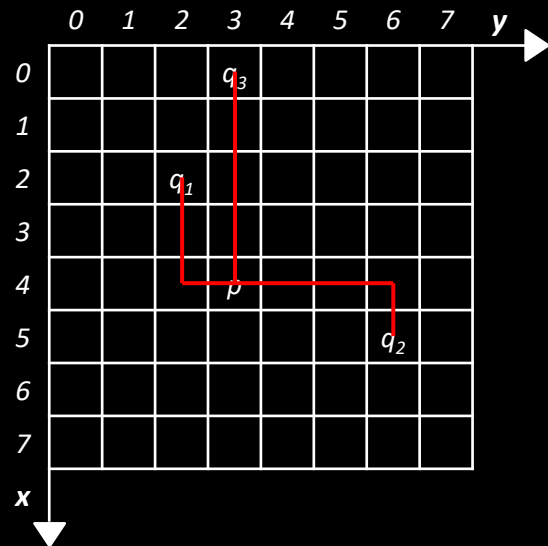
# Medidas de distância

- Distância *city block* entre  $p$  em  $(x, y)$  e  $q$  em  $(s, t)$ 
  - $D_4(p, q) = |x - s| + |y - t|$
- Para  $p$  com coordenadas  $(4, 3)$  e:
  - $q_1$  com coordenadas  $(2, 2)$ :
    - $D_4(p, q) = |4 - 2| + |3 - 2|$
    - $D_4(p, q) = 2 + 1 = 3$
  - $q_2$  com coordenadas  $(5, 6)$ :
    - $D_4(p, q) = |4 - 5| + |3 - 6|$
    - $D_4(p, q) = 1 + 3 = 4$
  - $q_3$  com coordenadas  $(0, 3)$ :
    - $D_4(p, q) = |4 - 0| + |3 - 3|$
    - $D_4(p, q) = 4 + 0 = 4$



# Medidas de distância

- Distância *city block* entre  $p$  em  $(x, y)$  e  $q$  em  $(s, t)$ 
  - $D_4(p, q) = |x - s| + |y - t|$
- Para  $p$  com coordenadas  $(4, 3)$  e:
  - $q_1$  com coordenadas  $(2, 2)$ :
    - $D_4(p, q) = |4 - 2| + |3 - 2|$
    - $D_4(p, q) = 2 + 1 = 3$
  - $q_2$  com coordenadas  $(5, 6)$ :
    - $D_4(p, q) = |4 - 5| + |3 - 6|$
    - $D_4(p, q) = 1 + 3 = 4$
  - $q_3$  com coordenadas  $(0, 3)$ :
    - $D_4(p, q) = |4 - 0| + |3 - 3|$
    - $D_4(p, q) = 4 + 0 = 4$



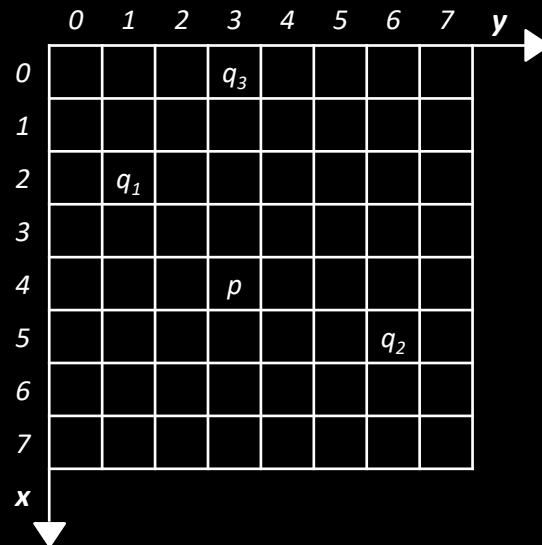
# Medidas de distância

- Distância *chessboard* entre  $p$  em  $(x, y)$  e  $q$  em  $(s, t)$ 
  - $D_8(p, q) = \max(|x - s|, |y - t|)$

	0	1	2	3	4	$y$
0	2	2	2	2	2	
1	2	1	1	1	2	
2	2	1	0	1	2	
3	2	1	1	1	2	
4	2	2	2	2	2	
$x$						

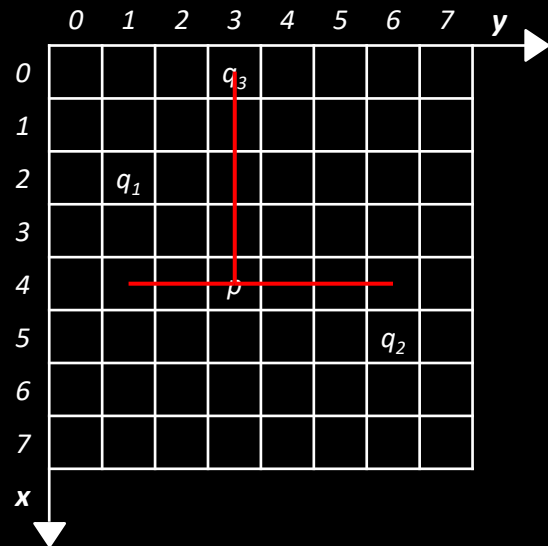
# Medidas de distância

- Distância *chessboard* entre  $p$  em  $(x, y)$  e  $q$  em  $(s, t)$ 
  - $D_8(p, q) = \max(|x - s|, |y - t|)$
- Para  $p$  com coordenadas  $(4, 3)$  e:
  - $q_1$  com coordenadas  $(2, 1)$ :
    - $D_8(p, q) = \max(|4 - 2|, |3 - 1|)$
    - $D_8(p, q) = \max(2, 2) = 2$
  - $q_2$  com coordenadas  $(5, 6)$ :
    - $D_8(p, q) = \max(|4 - 5|, |3 - 6|)$
    - $D_8(p, q) = \max(1, 3) = 3$
  - $q_3$  com coordenadas  $(0, 3)$ :
    - $D_8(p, q) = \max(|4 - 0|, |3 - 3|)$
    - $D_8(p, q) = \max(4, 0) = 4$



# Medidas de distância

- Distância *chessboard* entre  $p$  em  $(x, y)$  e  $q$  em  $(s, t)$ 
  - $D_8(p, q) = \max(|x - s|, |y - t|)$
- Para  $p$  com coordenadas  $(4, 3)$  e:
  - $q_1$  com coordenadas  $(2, 1)$ :
    - $D_8(p, q) = \max(|4 - 2|, |3 - 1|)$
    - $D_8(p, q) = \max(2, 2) = 2$
  - $q_2$  com coordenadas  $(5, 6)$ :
    - $D_8(p, q) = \max(|4 - 5|, |3 - 6|)$
    - $D_8(p, q) = \max(1, 3) = 3$
  - $q_3$  com coordenadas  $(0, 3)$ :
    - $D_8(p, q) = \max(|4 - 0|, |3 - 3|)$
    - $D_8(p, q) = \max(4, 0) = 4$



# Bibliografia

- MARQUES FILHO, O.; VIEIRA NETO, H. Processamento digital de imagens. Brasport, 1999.
  - Disponível para download no site do autor (Exclusivo para uso pessoal)
  - <http://dainf.ct.utfpr.edu.br/~hvieir/pub.html>
    - Seção 2.2 – pág. 25
    - Seção 2.3 – pág 28 (exceto Seção 2.3.3 )
  
- GONZALEZ, R.C.; WOODS, R.E.; Processamento Digital de Imagens. 3ª edição. Editora Pearson, 2009.
  - Disponível na Biblioteca Virtual da Pearson.
    - Seção 2.5 – pág. 44
    - Seção 2.6 – pág. 46 (até 2.6.4, parar antes de Conjuntos fuzzy (pág. 54)).
  
- J. E. R. Queiroz, H. M. Gomes. Introdução ao Processamento Digital de Imagens. RITA. v. 13, 2006.
  - <http://www.dsc.ufcg.edu.br/~hmg/disciplinas/graduacao/vc-2016.2/Rita-Tutorial-PDI.pdf>
  - Seção 3

# Bibliografia complementares

- scikit-image. **Image data types and what they mean.**
  - [https://scikit-image.org/docs/dev/user\\_guide/data\\_types.html](https://scikit-image.org/docs/dev/user_guide/data_types.html)
- Chityala, R; Pudipeddi, P. Image Processing and Acquisition using Python. CRC Press, 2014.



FIM