

9. a) Implemente duas versões deste algoritmo em qualquer linguagem de programação:

– versão 1: em tempo $\Omega(n \log n)$

```
def merge(A, p, q, r):
    L = [0] * (r + 1)
    for i in range(p, q + 1):
        L[i] = A[i]
    for j in range(q + 1, r + 1):
        L[r + q + 1 - j] = A[j]
    i = p
    j = r
    k = p
    for k in range(p, r + 1):
        if L[i] <= L[j]:
            A[k] = L[i]
            i += 1
        else:
            A[k] = L[j]
            j -= 1

def mergeSort(A, p, r):
    if p < r:
        q = (p + (r - 1)) // 2
        mergeSort(A, p, q)
        mergeSort(A, q + 1, r)
        merge(A, p, q, r)

def selectMedian(A, n):
    mergeSort(A, 0, n - 1)
    i = (n) // 2
    return A[i]
```

– versão 2: em tempo médio $O(n)$

```
def troca(A, i, j):
    aux = A[i]
    A[i] = A[j]
    A[j] = aux

def particione(A, p, r):
    pivo = A[r]
    i = p - 1
    for j in range(p, r):
        if A[j] <= pivo:
            i += 1
            troca(A, i, j)
    troca(A, i + 1, r)
    return (i + 1)

def particione_aleatorio(A, p, r):
    j = random.randrange(p, r)
    troca(A, j, r)
    return particione(A, p, r)

def select_NL(A, p, r, i):
    if p == r:
        return A[p]
    q = particione_aleatorio(A, p, r)
    k = q - p + 1
    if i == k:
        return A[q]
    elif i < k:
        return select_NL(A, p, q - 1, i)
    else:
        return select_NL(A, q + 1, r, i - k)
```

9. b) (OPCIONAL – pontuação extra) Implemente duas versões do filtro de mediana, considerando os dois algoritmos desenvolvidos no item (a), para matrizes bidimensionais $m \times n$ de inteiros $0 \leq f(i, j) \leq 255$, sendo $0 \leq i < m$, $0 \leq j < n$, supondo janela de filtro com vizinhança parametrizável de $p \times q$, sendo $2 \leq p < m$ e $2 \leq q < n$. A

técnica, exemplos e código (em C) podem ser consultados no seguinte documento:
<https://www.ime.usp.br/~reverbel/ccm118-12/eps/ep4.pdf>

Avalie o tempo de execução real (por exemplo, em segundos) das duas versões implementadas do filtro para uma matriz (imagem) suficientemente grande ($\geq 640 \times 480$ pixels) e para diferentes escolhas de p e q (por exemplo, 3, 7, 15, . . .).

Códigos:

```
def readImage (nome):
    imagem = cv2.imread(nome)
    gray_image = cv2.cvtColor(imagem, cv2.COLOR_BGR2GRAY)
    return gray_image

def aplica_janela(imagem, p, q, i, j, algoritmo_mediana):
    vetor = [0] * (p*q)
    k = 0
    if ((p % 2) == 0 or (q % 2) == 0):
        for r in range( i - (p/2), i + (p/2) - 1):
            for l in range(j - (q/2), j+(q/2) - 1):
                vetor[k] = imagem[r,l]
                k += 1
    else:
        for r in range( i - (p/2), i + (p/2)):
            for l in range(j - (q/2), j+(q/2)):
                vetor[k] = imagem[r,l]
                k += 1
    if (algoritmo_mediana == 1):
        a = select_NL(vetor, 0, k-1, math.floor((k+1)/2.0) )
    else:
        a = selectMedian(vetor, k, math.floor((k+1)/2.0))
    return a

def filtro_mediana(imagem, p, q, algoritmo_mediana):
    imagem_filtrada = copy.copy(imagem)
    if ( (p<2 or p >= imagem.shape[0]) or (q<2 or q >= imagem.shape[1]) ):
        print ("Tamanho invalido para janela de filtro!")
    else:
        for i in range(0,imagem.shape[0]):
            for j in range(0, imagem.shape[1]):
                if ( (i - (p/2) >= 0) and (j - (q/2) >= 0) and (i+(p/2) < imagem.shape[0]) and (j+(q/2) < imagem.shape[1]) ):
                    imagem_filtrada[i,j] = aplica_janela(imagem, p, q, i, j, algoritmo_mediana)
                else:
                    imagem_filtrada[i,j] = 0
    return imagem_filtrada
```

Imagem Original

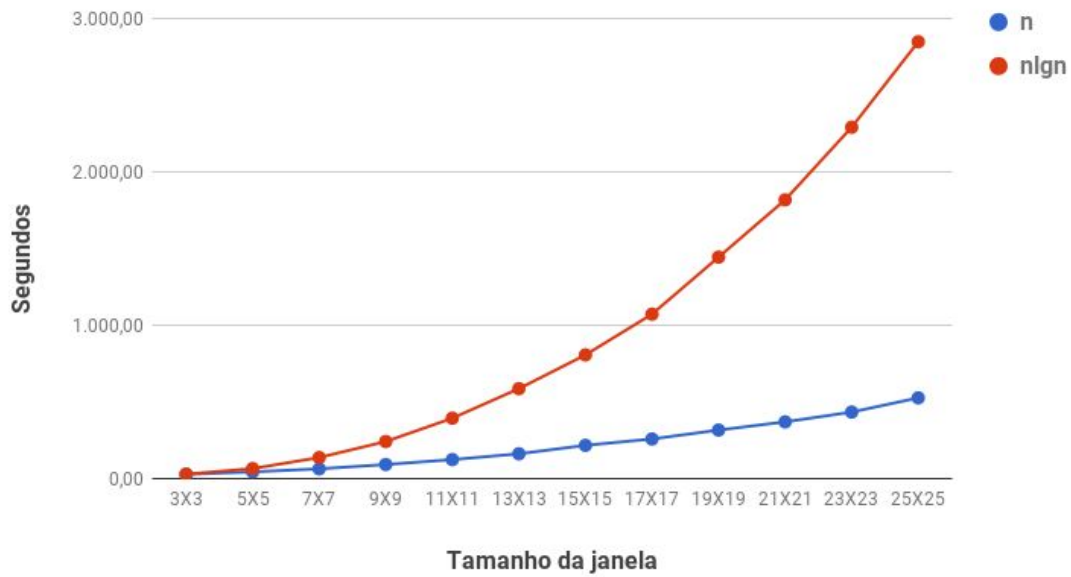
Nome: paisagem.jpg

Tamanho: 720 x 900



Resultado dos experimentos:

Desempenho das abordagens



A seguir são exibidos os resultados de cada experimento.

Experimento 1

IMAGEM: paisagem.jpg

IMAGEM: 720 X 900

Filtro: $O(n)$

- Tamanho da Janela: 3 X 3
- Tempo inicial: 1525797785.67
- Tempo final: 1525797810.05
- Tempo execução total: 24.3830571175



Experimento 2

IMAGEM: paisagem.jpg

IMAGEM: 720 X 900

Filtro: $O(n \log n)$

-- Tamanho da Janela: 3 X 3

-- Tempo inicial: 1525798525.17

-- Tempo final: 1525798550.4

-- Tempo execução total: 25.2275538445



Experimento 3

IMAGEM: paisagem.jpg

IMAGEM: 720 X 900

Filtro: $O(n)$

-- Tamanho da Janela: 5 X 5

-- Tempo inicial: 1525798656.22

-- Tempo final: 1525798696.67

-- Tempo execução total: 40.4442090988



Experimento 4

IMAGEM: paisagem.jpg

IMAGEM: 720 X 900

Filtro: $O(n \log n)$

-- Tamanho da Janela: 5 X 5

-- Tempo inicial: 1525798787.47

-- Tempo final: 1525798849.44

-- Tempo execução total: 61.9720978737



Experimento 5

IMAGEM: paisagem.jpg

IMAGEM: 720 X 900

Filtro: $O(n)$

-- Tamanho da Janela: 7 X 7

-- Tempo inicial: 1525799926.17

-- Tempo final: 1525799986.4

-- Tempo execução total: 60.2346711159



Experimento 6

IMAGEM: paisagem.jpg

IMAGEM: 720 X 900

Filtro: $O(n \log n)$

-- Tamanho da Janela: 7 X 7

-- Tempo inicial: 1525800037.88

-- Tempo final: 1525800172.0

-- Tempo execução total: 134.117254972



Experimento 7

IMAGEM: paisagem.jpg

IMAGEM: 720 X 900

Filtro: $O(n)$

-- Tamanho da Janela: 9 X 9

-- Tempo inicial: 1525800233.3

-- Tempo final: 1525800321.8

-- Tempo execução total: 88.5035991669



Experimento 8

IMAGEM: paisagem.jpg

IMAGEM: 720 X 900

Filtro: O(nlogn)

-- Tamanho da Janela: 9 X 9

-- Tempo inicial: 1525800355.24

-- Tempo final: 1525800594.04

-- Tempo execução total: 238.805572033



Experimento 9

IMAGEM: paisagem.jpg

IMAGEM: 720 X 900

Filtro: O(n)

-- Tamanho da Janela: 11 X 11

-- Tempo inicial: 1525800635.54

-- Tempo final: 1525800756.12

-- Tempo execução total: 120.586702108



Experimento 10

IMAGEM: paisagem.jpg

IMAGEM: 720 X 900

Filtro: $O(n \log n)$

-- Tamanho da Janela: 11 X 11

-- Tempo inicial: 1525800773.81

-- Tempo final: 1525801165.0

-- Tempo execução total: 391.187102079



Experimento 11

IMAGEM: paisagem.jpg

IMAGEM: 720 X 900

Filtro: $O(n)$

-- Tamanho da Janela: 13 X 13

-- Tempo inicial: 1525801231.84

-- Tempo final: 1525801390.38

-- Tempo execução total: 158.544473886



Experimento 12

IMAGEM: paisagem.jpg

IMAGEM: 720 X 900

Filtro: $O(n \log n)$

-- Tamanho da Janela: 13 X 13

-- Tempo inicial: 1525801452.18

-- Tempo final: 1525802036.3

-- Tempo execução total: 584.113286018



Experimento 13

IMAGEM: paisagem.jpg

IMAGEM: 720 X 900

Filtro: $O(n)$

-- Tamanho da Janela: 15 X 15

-- Tempo inicial: 1525802247.06

-- Tempo final: 1525802460.79

-- Tempo execução total: 213.731586933



Experimento 14

IMAGEM: paisagem.jpg

IMAGEM: 720 X 900

Filtro: $O(n \log n)$

-- Tamanho da Janela: 15 X 15

-- Tempo inicial: 1525802500.77

-- Tempo final: 1525803305.65

-- Tempo execução total: 804.878239155



Experimento 15

IMAGEM: paisagem.jpg

IMAGEM: 720 X 900

Filtro: $O(n)$

-- Tamanho da Janela: 17 X 17

-- Tempo inicial: 1525803354.67

-- Tempo final: 1525803610.05

-- Tempo execução total: 255.383650064



Experimento 16

IMAGEM: paisagem.jpg

IMAGEM: 720 X 900

Filtro: $O(n \log n)$

-- Tamanho da Janela: 17 X 17

-- Tempo inicial: 1525803643.58

-- Tempo final: 1525804714.53

-- Tempo execução total: 1070.95259213



Experimento 17

IMAGEM: paisagem.jpg

IMAGEM: 720 X 900

Filtro: $O(n)$

-- Tamanho da Janela: 19 X 19

-- Tempo inicial: 1525804789.2

-- Tempo final: 1525805102.74

-- Tempo execução total: 313.545382977



Experimento 18

IMAGEM: paisagem.jpg

IMAGEM: 720 X 900

Filtro: $O(n \log n)$

-- Tamanho da Janela: 19 X 19

-- Tempo inicial: 1525805127.55

-- Tempo final: 1525806571.39

-- Tempo execução total: 1443.84177804



Experimento 19

IMAGEM: paisagem.jpg

IMAGEM: 720 X 900

Filtro: $O(n)$

-- Tamanho da Janela: 21 X 21

-- Tempo inicial: 1525885454.78

-- Tempo final: 1525885821.87

-- Tempo execução total: 367.093325853



Experimento 20

IMAGEM: paisagem.jpg

IMAGEM: 720 X 900

Filtro: $O(n \log n)$

- Tamanho da Janela: 21 X 21
- Tempo inicial: 1525885862.28
- Tempo final: 1525887680.27
- Tempo execução total: 1817.99079394



Experimento 21

IMAGEM: paisagem.jpg

IMAGEM: 720 X 900

Filtro: $O(n)$

- Tamanho da Janela: 23 X 23
- Tempo inicial: 1525887702.08
- Tempo final: 1525888133.27
- Tempo execução total: 431.186913013



Experimento 22

IMAGEM: paisagem.jpg

IMAGEM: 720 X 900

Filtro: $O(n \log n)$

-- Tamanho da Janela: 23 X 23

-- Tempo inicial: 1525888152.84

-- Tempo final: 1525890445.56

-- Tempo execução total: 2292.71445107



Experimento 23

IMAGEM: paisagem.jpg

IMAGEM: 720 X 900

Filtro: $O(n)$

-- Tamanho da Janela: 25 X 25

-- Tempo inicial: 1525806719.54

-- Tempo final: 1525807243.69

-- Tempo execução total: 524.145744085



Experimento 24

IMAGEM: paisagem.jpg

IMAGEM: 720 X 900

Filtro: $O(n \log n)$

-- Tamanho da Janela: 25 X 25

-- Tempo inicial: 1525807280.03

-- Tempo final: 1525810131.36

-- Tempo execução total: 2851.33629513

