

## Lista de exercícios

1) Implemente o algoritmo Merge-Sort visto em aula, e para tanto, codifique sem consultar a internet a função **Intercala**. A implementação do algoritmo mais difundida utiliza um vetor auxiliar que ajudar a armazenar os elementos intercalados. Teste e compute o tempo do algoritmo para 100, 1.000, 10.000, 100.000, 1.000.000 elementos (gerados de forma aleatória, em ordem crescente e em ordem decrescente). Utilize o programa **merge\_sort.c** em anexo ao material da aula.

2) A filtragem de imagens visando a redução do ruído é uma tarefa muito importante em processamento de imagens, e encontra diversas aplicações nas áreas de sensoriamento remoto, inspeção industrial e na medicina. O filtro da mediana é uma transformação bastante comum para suavizar ruídos do tipo impulsivo (sal-e-pimenta) em sinais e imagens digitais.

Antes de definir o filtro da mediana para imagens vamos definir o conceito de vizinhança: seja  $M$  uma matriz de inteiros positivos com  $m$  linhas e  $n$  colunas, e seja  $p$  um inteiro positivo ímpar. Dada uma coordenada  $(i, j)$  em  $M$ , a vizinhança de tamanho  $p \times p$  em torno de  $(i, j)$  é a submatriz  $M_{i,j}$  de  $M$  com  $p$  linhas e  $p$  colunas e centro em  $(i, j)$ . Por exemplo, dada a seguinte matriz  $5 \times 5$

$$M = \begin{pmatrix} 10 & 18 & 21 & 0 & 8 \\ 1 & 7 & 2 & 7 & 1 \\ 0 & 9 & 34 & 23 & 12 \\ 16 & 18 & 6 & 15 & 19 \\ 1 & 18 & 3 & 23 & 4 \end{pmatrix} \quad (1)$$

a vizinhança  $3 \times 3$  em torno da coordenada  $(1, 1)$  é a submatriz

$$M_{1,1} = \begin{pmatrix} 10 & 18 & 21 \\ 1 & 7 & 2 \\ 0 & 9 & 34 \end{pmatrix} \quad (2)$$

Note que a vizinhança não é bem definida em algumas coordenadas (por exemplo, em  $(0, 0)$ ).

Uma imagem digital pode ser representada por uma matriz. Dada uma matriz  $M$  de inteiros positivos com  $m$  linhas e  $n$  colunas, e um inteiro positivo ímpar  $p$ , o filtro da mediana calcula uma matriz  $F$  com o mesmo tamanho de  $M$ , de forma que  $F(i, j)$  contém a mediana dos números em  $M(i, j)$  (a vizinhança  $p \times p$  em torno de  $(i, j)$ ).

No caso do exemplo anterior, os números em torno de  $(1, 1)$  são 10, 18, 21, 1, 7, 2, 0, 9, 34. Logo,  $F(1, 1) = 9$  (valor da mediana em 0, 1, 2, 7, 9, 10, 18, 21, 34). Quando a vizinhança de uma coordenada  $(i, j)$  não estiver bem definida, usaremos a convenção  $F(i, j) = 0$ .

No caso da matriz-exemplo da seção anterior, o resultado do filtro da mediana com uma vizinhança de  $3 \times 3$  é a seguinte matriz:

$$F = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 9 & 9 & 12 & 0 \\ 0 & 7 & 9 & 12 & 0 \\ 0 & 9 & 18 & 15 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad (3)$$

Neste exercício utilizaremos o formato PGM (portable graymap) para armazenar imagens em arquivos. Este formato tem duas variações, uma binária (o PGM “normal” ou raw) e outra textual (o PGM ASCII ou plain). Em ambos os casos, o arquivo deve conter um cabeçalho e a matriz correspondente à imagem. O exemplo a seguir mostra um arquivo PGM textual:

```
P2
5 4
34
10 18 21 0 8
1 7 2 7 1
0 9 34 23 12
16 18 6 15 19
```

A primeira linha do arquivo contém obrigatoriamente uma palavra-chave, no caso desse exercício ela deve ser P2 (arquivo PGM textual). A segunda linha contém dois números inteiros que indicam o número de colunas e o número de linhas da matriz, respectivamente. A terceira linha contém um número inteiro positivo maxval, que deve ser igual ao maior elemento da matriz. Na definição do formato PGM, maxval não pode ser maior que 65535. Para fins deste exercício, entretanto, maxval é no máximo 255. Os demais números do arquivo são os elementos de uma matriz de inteiros com os tons de cinza de cada ponto da imagem. Cada tom de cinza é um número entre 0 e maxval, com 0 indicando “preto” e maxval indicando “branco”.

Existem dois outros formatos de arquivo muito semelhantes ao PGM: o PBM (portable bitmap), para imagens monocromáticas (só preto e branco, sem tons de cinza), e o PPM (portable pixmap), para imagens coloridas. Quem quiser saber mais detalhes sobre esses formatos, visite as seguintes páginas:

- [http://en.wikipedia.org/wiki/Portable\\_pixmap](http://en.wikipedia.org/wiki/Portable_pixmap)

Para quem usa o sistema operacional Windows use o programa IrfawView para visualizar as imagens de exemplo.

Neste exercício você deve implementar um programa simples para filtrar ruídos tipo sal-e-pimenta em uma imagem. Seu programa deve permitir que o usuário abra um arquivo contendo uma imagem, aplique o filtro da mediana sobre a imagem e salve a imagem modificada em um arquivo de saída. O programa irá trabalhar apenas com imagens cujos pontos podem assumir tonalidades de cinza que variam do branco ao preto (PGM).

Avalie o efeito ao utilizar diversos tamanhos de vizinhança para filtragem,  $5 \times 5$ ,  $9 \times 9$ ,  $17 \times 17$ , ... A mediana deve ser calculada por algum algoritmo de ordenação visto em aula.

O diretório “imagens” contém 3 exemplos de imagens com ruído sal-e-pimenta teste seu algoritmo com todas elas.

Utilize, caso queira, o código **imagens.c** em anexo ao material da aula (que já realiza a leitura das imagens).