

ACH2002 - Introdução à Análise de Algoritmos (2º. Semestre/2024)

Prof.Dra. [Ariane Machado Lima](#)

Exercício de Programação 1: **Recursão/Divisão e Conquista**

Data de entrega: **22/09/2024 - domingo (até 23:59h)**

1. Objetivo

Desenvolver um programa em linguagem C que implemente um algoritmo **obrigatoriamente** utilizando **recursão e divisão e conquista** para identificar a coordenada e o valor do pixel mais escuro (menor valor) e a coordenada e o valor do pixel mais claro (maior valor) em uma imagem digital em nível de cinza representada em uma matriz quadrada $M_{n \times n}$ sendo n uma potência de 2.

2. Definições teóricas

Uma **matriz 2-D** (bidimensional) é uma forma utilizada para representar quantitativamente uma imagem digital. Cada elemento da matriz é chamado *elemento de imagem, elemento pictórico ou pixel*.

A **figura 1** mostra uma ampliação de uma imagem digital de oito por oito pixels (**8x8**), contendo um total de **64 pixels**. A altura em pixels dessa imagem é 8, e a largura em pixels dessa imagem é 8. Uma matriz 2-D quadrada que representa essa imagem contém 8 linhas e 8 colunas.

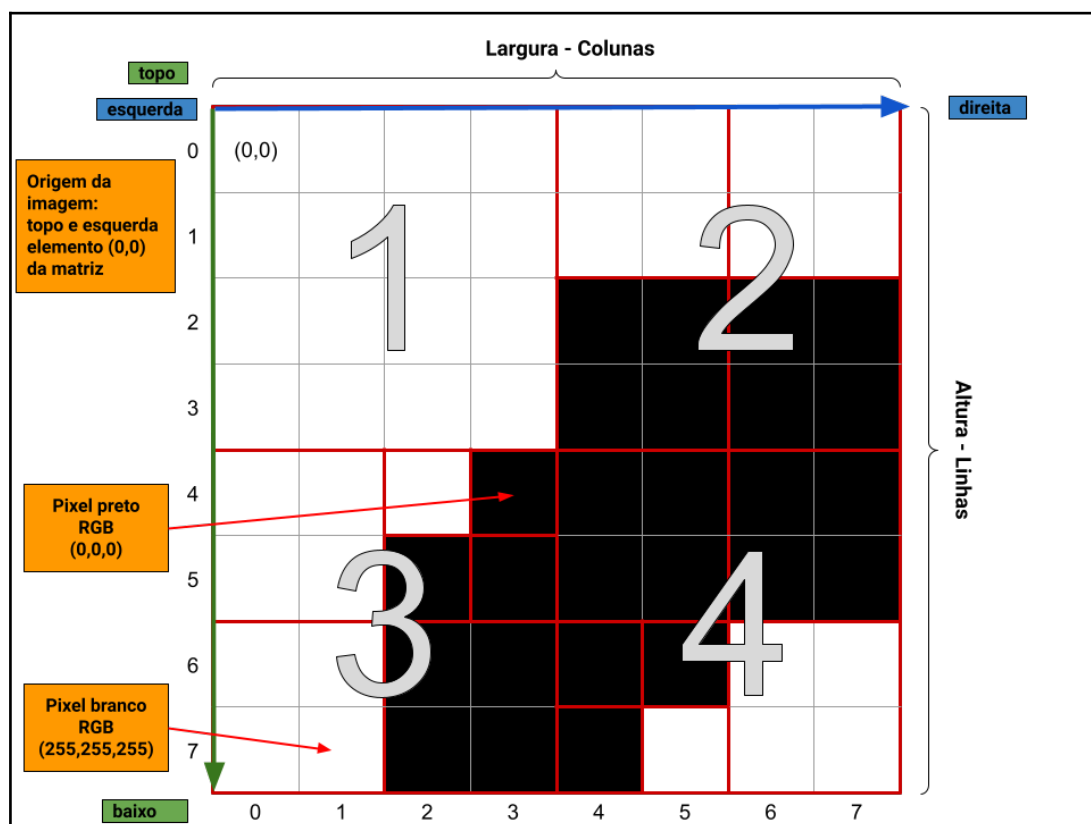


Figura 1 - Representação da ampliação de uma imagem digital 8x8.

Na **Figura 1**, os números grandes 1, 2, 3 e 4 identificam quatro quadrantes da imagem, cada um com dimensão 4 (4x4). Cada quadrante é dividido em 4 subquadrantes, e assim sucessivamente, até chegar ao pixel. A imagem da **Figura 1** contém: um quadrante **8x8** (a própria imagem), quatro quadrantes **4x4**, 16 quadrantes **2x2**, e finalmente, 64 quadrantes **1x1** (cada pixel).

Em uma imagem em nível de cinzas, cada pixel contém um valor de 0 a MAX, sendo que MAX depende do número de níveis de cinza considerados. Neste exercício-programa consideraremos MAX = 255. Logo, uma imagem contendo NxN pixels pode ser convertida em uma matriz NxN de valores 0 (**pixels PRETOS absolutos**) a 255 (**pixels BRANCOS absolutos**).

3. Definições práticas

- A representação da imagem digital será uma **matriz quadrada NxN**, onde **N será uma potência de 2 e no máximo 64**;
- O programa **deve obrigatoriamente** aceitar três parâmetros de entrada: **(1)** um inteiro **N** (potência de 2 e no máximo 64, a dimensão da matriz que representa a imagem); **(2)** o nome do arquivo de entrada; **(3)** o nome do arquivo de saída.
- Exemplo (execução no linux): considerando que **ep1** é o nome do programa executável:
 - `$/ep1 8 minha_entrada.txt minha_saida.txt`
- O arquivo de entrada será a representação textual dos valores dos pixels de uma imagem em 256 níveis de cinza, ou seja, o arquivo de entrada conterá **N** linhas, cada uma contendo **N** números (0 a 255) com pelo menos um espaço separando cada um deles (sem espaço no final da linha). **Dica:** use `fscanf` para ler o arquivo de entrada.
- A **alocação** da matriz poderá ser **dinâmica ou estática**;
- O programa fonte em linguagem C **deverá** consistir de um único arquivo chamado `ep1_XXXX.c` sendo XXXX o número USP do aluno;
- O programa em C deve ser compilável em **gcc** utilizando linha de comando no *prompt* de um terminal Linux¹; **qualquer opção de compilação deve ser explicitamente informada na entrega**;
- O programa deve gerar um arquivo de saída com nome exatamente igual ao terceiro parâmetro da chamada do programa, no mesmo diretório onde o programa é executado.

4. Definição do arquivo de saída

O arquivo de saída deverá conter duas linhas exatamente como descrito abaixo (**repetindo espaços** representados com fundo cinza e **letras em minúsculas**):

min = <menor_valor>; pos = (<i_min>, <j_min>)
max = <maior_valor>; pos = (<i_max>, <j_max>)

¹ No Windows você pode testar usando o cmd.exe, porém só vou testar no Linux.

em que:

<menor_valor> é o menor valor de pixel da matriz inteira;

<i_min> e **<j_min>** são, respectivamente o número da linha e da coluna em que o menor valor está presente;

<maior_valor> é o maior valor de pixel da matriz inteira;

<i_max> e **<j_max>** são, respectivamente o número da linha e da coluna em que o maior valor está presente.

Caso duas coordenadas distintas contiverem o mesmo valor mínimo (ou máximo), deve ser impressa a coordenada com o menor valor da linha, e em caso de empate, a que tiver o menor valor de coluna. Ex:

- se o valor mínimo (ou máximo) ocorrer nas coordenadas (3, 7) e (6, 3), deve ser impressa a coordenada (3, 7);
- se o valor mínimo (ou máximo) ocorrer nas coordenadas (3, 7) e (3, 3), deve ser impressa a coordenada (3, 3);

Dica: use `fprintf` para escrever no arquivo de saída.

Exemplo de arquivo de **entrada**, para N=4

```
7 89 250 68
126 223 109 45
87 126 62 253
78 201 7 75
```

O arquivo de **saída** deverá ser (**note as minúsculas e os espaços de acordo com a definição acima!!!!**):

```
min = 7; pos = (0, 0)
max = 253; pos = (2, 3)
```

5. Orientações

- O trabalho é **INDIVIDUAL**, com o nome do aluno e número USP devidamente identificado nas primeiras linhas do código-fonte.
- A entrega será realizada em atividade específica para o EP1 no ambiente e-disciplinas, até a data marcada no cabeçalho deste arquivo.
- Deverá ser entregue:
 - O arquivo com o programa-fonte em C.
 - O programa será compilado como abaixo:
■ `$gcc -o ep1 ep1_XXXX.c`
- Trabalhos com evidências de plágio de qualquer tipo (mesmo que parcial) serão desconsiderados e os autores receberão nota **ZERO**.