

Problema: Identificando a Região de Interesse de uma Imagem

1. Background

Imagine que você trabalha com processamento de imagens e precisa identificar Regiões de Interesse (do inglês ROI – *Region of Interest*) em imagens específicas. A região de interesse de uma imagem é aquela em que maior parte do processamento será realizada. No caso da aplicação de um filtro, Sobel¹ por exemplo, é a região onde encontramos mais bordas, indicando que há uma concentração maior de objetos nesta região.

Veja o exemplo abaixo:



Fonte: https://en.wikipedia.org/wiki/Sobel_operator

Uma imagem digital, nada mais é do que uma matriz de pixels que assumem valores diferentes fornecendo informações sobre as cores da imagem.

¹ Sobel filter - https://en.wikipedia.org/wiki/Sobel_operator

2. Definição do problema

Para fins de desenvolvimento do trabalho, abordaremos o problema de forma simples: dado um array bidimensional (matriz) de inteiros positivos e negativos, sua tarefa é encontrar o subretângulo com a maior soma dentro da matriz (representando a Região de Interesse da imagem). A soma de um retângulo é a soma de todos os elementos naquele retângulo. Neste problema, o subretângulo

com a maior soma é referido como *subretângulo máximo*. Um subretângulo é qualquer subarray contíguo de tamanho 1×1 ou maior localizado dentro do array total. Como exemplo, o subretângulo máximo do array:

0	0	1	0
9	2	-6	2
-3	2	-4	1
-2	6	0	-3

está localizado na parte inferior esquerda (9 – 2; -3 – 2; -2 – 6) e sua soma total é 14.

Entrada e Saída

A entrada consiste em um array de inteiros de tamanho $N \times N$. A entrada começa com um inteiro positivo N na primeira linha indicando o tamanho da matriz quadrada. As linhas subsequentes apresentam os inteiros de cada linha da matriz separados por espaço (cada linha da matriz será apresentada em uma linha diferente da entrada). O maior valor assumido por N será 100. Os inteiros presentes no array serão compostos por valores no intervalo $[-127, 127]$.

A saída é a soma do *subretângulo máximo*.

Exemplo de entrada

```
4
0 0 1 0
9 2 -6 2
-3 2 -4 1
-2 6 0 -3
```

Exemplo de saída

```
14
```

3. Orientações para desenvolvimento do trabalho

O trabalho deverá ser desenvolvido **em dupla**. Cada dupla deverá entregar um artigo e apresentar a solução para a professora.

- O artigo deve:
 - Apresentar o(s) algoritmo(s) solução para o problema.
 - Apresentar gráfico(s) e/ou tabela(s) com tempo de execução.
 - Apresentar a análise do(s) algoritmo(s) proposto(s).
 - Possuir no máximo 4 páginas.

- Seguir o modelo de artigos da SBC (Sociedade Brasileira de Computação).
- Estar bem escrito. Trabalhos mal escritos terão nota compatível.
- A apresentação será feita por meio de perguntas e respostas: a professora pergunta, e a dupla responde.
- Saber responder as perguntas faz parte da avaliação.

4. Orientações para a entrega

- O artigo deve estar em formato pdf.
- A(s) implementação(ões) das soluções propostas devem ser entregues e conter arquivo README especificando dependências de bibliotecas externas, além do processo de compilação e execução. Serão aceitos trabalhos implementados em C/C++, Java e Python.
- Para entrega, compactar todos os arquivos em um único **em formato .zip**.
- Prazo para entrega: até **18h de 17/11/2025**.

IMPORTANTE:

- Não serão aceitos trabalhos em atraso.
- O tempo de apresentação será definido de acordo com a quantidade de duplas.
- A ordem das apresentações será sorteada pela professora.

5. Avaliação

A avaliação contemplará a(s) solução(ões) proposta(s), o artigo e a apresentação. Os critérios serão:

- a) originalidade da solução principal – 3.0 pontos
- b) apresentação de soluções alternativas (pelo menos uma) – 2.0 pontos
- c) artigo – 3.0 pontos
- d) apresentação clara e correta – 2.0 pontos

Cabe salientar que a reprodução (praticamente) literal de conteúdo produzido por outro autor é considerada **plágio** (olhar aqui: <http://www.unisinos.br/plagio>). Na detecção de plágio, a nota será automaticamente **zerada**.