

## Algoritmos e Estruturas de Dados

2024/2025 — 1º Semestre

### 1º Trabalho — O TAD imageBW

Data limite de entrega: 2 de dezembro de 2024

Este trabalho tem como objetivos principais:

1. Efectuar o **desenvolvimento e teste do TAD imageBW**, que permite instanciar e operar com **imagens binárias** (BW – *black-and-white*), em que cada *pixel* pode tomar um valor de intensidade **0** (*white*) ou **1** (*black*).

Cada imagem é representada internamente usando uma estratégia simples de compressão sem perdas: **RLE – Run-Length-Encoding**.

Usando RLE, uma sequência de *pixels* adjacentes com o mesmo valor de intensidade (*run*) é representada de modo compacto, indicando quantos *pixels* a constituem e qual o seu valor de intensidade. E duas *runs* adjacentes têm valores de intensidade distintos.

Para imagens *black-and-white*, é suficiente indicar o valor de intensidade do 1º pixel de cada linha da imagem, não sendo necessário indicar os valores de intensidade das subsequentes *runs* de uma mesma linha, uma vez que terão valores de intensidade alternados.

Assim, cada linha de uma imagem *black-and-white* é representada por um *array* em que o 1º elemento contém o valor de intensidade do 1º pixel dessa linha, seguindo-se vários elementos contendo o número de *pixels* constituindo cada *run*; o último elemento do *array* será um elemento terminador ( -1), indicando o final dessa linha da imagem.

#### Exemplos simples:

```
// Linha com pixels de intensidade constante constitui uma só run
// É representada por um array com apenas 3 elementos
1 1 1 1 -> 1 4 -1
```

```
// Linha com n pixels de intensidade alternada é constituída por n runs
// É representada por um array com (n + 2) elementos
0 1 0 1 -> 0 1 1 1 1 -1
```

2. Analisar, para a função **ImageCreateChessboard(...)**, o espaço de memória ocupado pela **imagem criada**, em função do seu número de linhas (**m**), de colunas (**n**) e do comprimento da aresta (**s**) de cada quadrado do padrão de xadrez criado.
3. Analisar a **complexidade computacional** da função **ImageAND (...)**.

## 1 – Desenvolver o TAD imageBW

Com base nos ficheiros fornecidos **imageBW.h** e **imageBW.c**, deve:

1. **Analisar** com cuidado os dois ficheiros.
2. **Concluir o desenvolvimento** das funções especificadas no ficheiro de interface, mas que ainda não estejam concluídas no ficheiro de implementação.
3. **Testar** todas as funções desenvolvidas, usando os programas de teste fornecidos.

### Atenção:

- O ficheiro de interface **imageBW.h** especifica as diferentes funções do TAD e **não deve ser alterado**.
- Para cada função são indicadas as eventuais pré-condições e pós-condições, que deverão ser obrigatoriamente tidas em conta.
- Caso facilite o desenvolvimento do código, poderão ser definidas e implementadas funções auxiliares adicionais (**static**) no ficheiro **imageBW.c**.
- O **código** desenvolvido deverá ser **claro** e **comentado** de modo apropriado: os identificadores escolhidos para as variáveis e a estrutura do código, bem como os eventuais comentários, deverão ser suficientes para a sua compreensão.
- Não é necessário entregar qualquer relatório relativo ao desenvolvimento do TAD.
- Deverá apenas ser entregue o ficheiro **imageBW.c** — com a **identificação do(s) seu(s) autor(es)**.

## 2 – Para a função **ImageCreateChessboard(...)**, analisar o espaço de memória ocupado pela imagem criada

Deve analisar o espaço de memória ocupado pelas imagens criadas pela função **ImageCreateChessboard()** em função do seu número de linhas (**m**), de colunas (**n**) e do comprimento da aresta (**s**) de cada quadrado do padrão de xadrez criado.

1. Crie imagens com vários tamanhos e diferentes padrões de xadrez. Registe as suas características e o espaço de memória total ocupado. Obtenha **expressões** para o número de *runs* e para o espaço de memória ocupado.
2. Qual é o padrão de xadrez que dá origem à **imagem com menos runs**? Quantas são? Qual é o padrão de xadrez que dá origem à **imagem com mais runs**? Quantas são?

## 3 – Analisar a complexidade computacional da função **ImageAND(...)**

Deve analisar a eficiência computacional do algoritmo desenvolvido para a função **ImageAnd()**.

1. Realize testes computacionais com imagens de diferentes tamanhos. Registe e analise o número de **operações** efectuadas envolvendo os valores dos *pixels* das imagens.

2. Efetue uma **análise formal** da complexidade do algoritmo.
3. Compare os resultados obtidos nas duas tarefas anteriores.

#### 4 – Escrever um relatório sucinto (máx. 6 págs.). O relatório deverá incluir:

Para cada uma das funções analisadas:

- i) Uma tabela com os resultados dos testes efectuados.
- ii) A análise formal da complexidade das funções em causa.

Para a função `ImageAnd()` será **valorizada** a comparação de **diferentes estratégias** algorítmicas para essa operação.

#### 5 – Critérios de Avaliação

- **Colaboração na Avaliação entre Pares** (10%)
  - Testar o funcionamento do código de colegas, avaliar a sua qualidade e clareza (2 trabalhos)
- **Desenvolvimento e teste das funções pedidas** (40%)
  - Será avaliada a qualidade e clareza do código e comentários
  - Será verificada a existência de eventuais fugas de memória
- **Relatório:** (50%)
  - Aspectos Gerais/Apresentação/Conclusão
  - Análise da função `ImageCreateChessboard()`
    - Dados experimentais
    - Análise do espaço de memória ocupado pela imagem criada
  - Análise da função `ImageAnd()`
    - Dados experimentais
    - Análise Formal
    - Análise Comparativa: Algoritmo Básico/Algoritmo Melhorado

A **nota atribuída ao desenvolvimento e teste das funções pedidas** será obtida pela média da classificação atribuída pelo docente e da classificação atribuída pelos colegas (avaliação entre pares, cada trabalho será avaliado por 3 a 4 estudantes):

A **avaliação entre pares está sujeita a validação pelos docentes**, podendo ser descartada nos casos em que se verifique que corresponde a uma avaliação manifestamente incorreta do trabalho apresentado.

#### Atenção:

- O trabalho deve ser realizado em grupos de 2 alunos.
- Serão disponibilizados ficheiros adicionais na plataforma **GitHub Classroom**, onde poderão testar o funcionamento correto das funções à medida que forem sendo desenvolvidas.

- A entrega do trabalho (**ficheiro imageBW.c + relatório**) será feita através da plataforma eLearning.
- Após a submissão dos trabalhos cada aluno receberá informação sobre os dois trabalhos que deverá rever. Esta tarefa é parte integrante deste trabalho.