

**Projeto de Avaliação**  
**Estruturas de Dados Avançadas (EDA)**  
**EST-IPCA**

**Barcelos**  
**23 de fevereiro de 2024**

### **Motivação**

Este projeto de avaliação de realização individual da Unidade Curricular (UC) *Estruturas de Dados Avançadas* (EDA), integrada no 2º semestre do 1º ano, visa o reforço e a aplicação dos conhecimentos adquiridos ao longo do semestre.

Com este projeto de avaliação pretende-se sedimentar os conhecimentos relativos à definição e manipulação de estruturas de dados dinâmicas na linguagem de programação C.

### **Objetivo**

Este documento deve ser visto como uma referência para uma abordagem clássica de desenvolvimento de soluções de software para um problema de dimensão média. A implementação das soluções deverá considerar estruturas de dados dinâmicas, armazenamento em ficheiro, modularização e apresentar uma estruturação e documentação com Doxygen.

### **Fase 1 - Listas ligadas**

Pretende-se uma solução para o cálculo da soma máxima possível dos inteiros de uma matriz de inteiros com qualquer dimensão, de modo que nenhum dos inteiros selecionados compartilhe a mesma linha ou coluna.

Por exemplo, a soma máxima possível para a matriz abaixo é igual a 3315 (sendo que os inteiros selecionados são 863, 383, 343, 959 e 767):

7	53	183	439	<b>863</b>
497	<b>383</b>	563	79	973
287	63	<b>343</b>	169	583
627	343	773	<b>959</b>	943
<b>767</b>	473	103	699	303

Procure implementar as funcionalidades seguintes:

1. Definição de uma estrutura de dados dinâmica *ED*, recorrendo à definição de listas ligadas, para a representação de uma matriz de valores inteiros;
2. Carregamento para a estrutura de dados da alínea anterior *ED* dos dados de uma matriz constante num ficheiro de texto. A operação deverá considerar matrizes com qualquer dimensão, sendo os valores separados por vírgulas. A título de exemplo, o ficheiro de texto deverá respeitar o formato seguinte:

```
7;53;183;439;863
497;383;563;79;973
287;63;343;169;583
627;343;773;959;943
767;473;103;699;303
```

3. Alteração dos inteiros constantes em *ED*;
4. Inserção de novas linhas/colunas na matriz representada por *ED*;
5. Remoção de linhas/colunas na matriz representada por *ED*;
6. Listagem de forma tabular na consola de todos os inteiros constantes na estrutura de dados *ED*;
7. Cálculo da soma máxima possível dos inteiros em *ED* de modo que nenhum dos inteiros selecionados compartilhe a mesma linha ou coluna.

Gravar a matriz no ficheiro

## Fase 2 - Grafos

Nesta segunda fase, pretende-se aplicar conceitos avançados de teoria dos grafos e programação em C para resolver um problema computacional com grau de complexidade maior, relacionando estruturas de dados, algoritmos de procura e técnicas de otimização. O objetivo é desenvolver uma solução capaz de calcular o somatório máximo possível de inteiros a partir de uma matriz de inteiros de dimensões arbitrárias, considerando regras específicas de conexão entre os inteiros.

Assim, procure implementar as funcionalidades seguintes:

1. Definir uma estrutura de dados *GR* para representar um grafo. Esta estrutura deve ser capaz de representar grafos dirigidos e deve suportar um número variável de vértices. A implementação deve incluir funções básicas para criação do grafo, adição e remoção de vértices e arestas;

2. Após definir a estrutura de dados GR, pretende-se modelar o problema utilizando grafos. Cada elemento da matriz de inteiros será representado por um vértice no grafo. As arestas entre vértices devem representar a possibilidade de somar dois elementos adjacentes na matriz, sob uma regra de conexão específica que poderá ser configurada pelo utilizador (por exemplo, apenas elementos na mesma linha ou coluna, não permitindo diagonais, ou qualquer outra regra);
8. Carregamento para a estrutura de dados da alínea anterior GR dos dados de uma matriz de inteiros constante num ficheiro de texto. A operação deverá considerar matrizes de inteiros com qualquer dimensão, sendo os valores separados por vírgulas. A título de exemplo, o ficheiro de texto deverá respeitar o formato seguinte:

```
7;53;183;439;863
497;383;563;79;973
287;63;343;169;583
627;343;773;959;943
767;473;103;699;303
```

3. Implementar operações de manipulação de grafos, incluindo procura em profundidade ou em largura, para identificar todos os caminhos possíveis que atendem às regras de conexão definidas. Desenvolver também uma função para calcular a soma dos valores dos vértices num dado caminho;
4. Utilizar as estruturas e algoritmos desenvolvidos para encontrar o caminho que proporciona a maior soma possível dos inteiros na estrutura GR, seguindo a regra de conexão estabelecida. O programa deve fornecer tanto a soma máxima quanto o caminho (ou caminhos, se existirem múltiplos caminhos com a mesma soma máxima) que resulta nessa soma;

Documente extensivamente o código desenvolvido, explicando a lógica por trás das principais funções e decisões de implementação. Execute casos de teste com matrizes de diferentes dimensões e complexidades, demonstrando a eficiência da solução desenvolvida.

## **Entrega**

A submissão de cada projeto deverá ser efetuada através da página da unidade curricular na plataforma *Moodle* em conformidade com as datas seguintes:

Fase 1: até dia 16 de março de 2024;

Fase 2: até dia 25 de maio de 2024.

Deverá ser submetido um ficheiro ZIP com todo o material desenvolvido (código e documentação).

**Defesa de cada fase**

A defesa de cada fase é obrigatória, realizando-se em data a divulgar posteriormente.

**Critérios de Avaliação**

Os critérios de avaliação incorporam os pontos seguintes:

- Qualidade do código desenvolvido (15%);
- Qualidade da solução desenvolvida (15%);
- Qualidade da documentação produzida (10%);
- Qualidade da defesa do projeto desenvolvido (60%);