

LESI 2023/24

# Estrutura de Dados Avançadas

Projeto de Avaliação - Parte II

Miguel Pires - 27968  
25-05-2024

## Índice

Introdução.....	3
Desenvolvimento.....	4
Análise e Modelação .....	4
Tipo de Grafo .....	4
Estrutura .....	5
Estrutura para Ficheiros .....	6
Representação dos dados no Grafo (Adjacências).....	7
Implementação.....	8
Contar Ficheiro .....	8
Importação de Adjacências.....	9
Análise de Resultados .....	10
Operações não implementadas .....	10
Abordagens Exploradas .....	10
Conclusão.....	11
Trabalho a Desenvolver.....	11

## Índice de Figuras

Figura 1 - Estrutura Vértice .....	5
Figura 2 - Estrutura Adjacência .....	5
Figura 3 - Estrutura Grafo.....	5
Figura 4 - Estrutura para Ficheiros.....	6
Figura 5 - Representação dos Pesos entre Vértices .....	7
Figura 6 - Função ContarFicheiro .....	8
Figura 7 - Função ImportaFicheiro .....	9

## Introdução

O presente trabalho tem como objetivo aplicar conceitos avançados de teoria dos grafos e programação em C para a resolução de problemas. O problema em questão envolve a construção de uma estrutura de dados eficiente para representar grafos, a fim de modelar e otimizar a soma máxima de inteiros.

A proposta de trabalho consiste em desenvolver um programa que seja capaz de representar grafos orientados e pesados capazes de suportar um número indeterminado de vértices (dinâmicos), bem como implementar as funções básicas para criação, remoção e inserção de vértices e adjacências, e posteriormente funções mais avançadas como identificar caminhos e calcular somas dos caminhos a percorrer.

# Desenvolvimento

## Análise e Modelação

A estrutura de dados fornecida é fundamental para a representação e manipulação de grafos orientados, permitindo a modelagem de grafos de maneira eficiente e flexível.

### Tipo de Grafo

O grafo utilizado, segue uma estrutura orientada dinâmica com suporte para múltiplos vértices e adjacências. Contém uma abordagem de listas encadeadas para armazenar vértices e as respectivas adjacências que facilita a adição e remoção dinâmica de elementos, bem como a implementação de algoritmos.

## Estrutura

O conjunto destas estruturas (Vértice, Adjacente e Grafo), permite a implementação de um grafo dinâmico orientado.

```
typedef struct Vertice {  
    int id;  
    int valor;  
    bool percorrido;  
    struct Adjacente* proxAdjacente;  
    struct Vertice* proxVertice;  
}Vertice;
```

Figura 1 - Estrutura Vértice

```
typedef struct Adjacente {  
    int id;  
    int peso;  
    struct Adjacente* prox;  
}Adjacente;
```

Figura 2 - Estrutura Adjacência

```
typedef struct Grafo {  
    struct Vertice* grafoLV;  
    int numeroVertices;  
    int totalVertices;  
}Grafo;
```

Figura 3 - Estrutura Grafo

## Estrutura para Ficheiros

Estrutura que permite guardar o grafo num ficheiro.

```
// Guardar Ficheiro
typedef struct FicheiroAdj {
    int idOF;
    int idDF;
    int pesoF;
}FicheiroAdj;

typedef struct FicheiroVertice {
    int idF;
    int valorF;
} FicheiroVertice;
```

Figura 4 - Estrutura para Ficheiros

Foi necessário a criação de novas estruturas específicas para guardar os ficheiros, pois como o valor dos apontadores muda a cada execução, não há motivo para guardar os apontadores.

## Representação dos dados no Grafo(Adjacências)

O ficheiro carregado “Pesos.txt”, foi utilizado como o peso entre adjacências:

	V0	V1	V2	V3	V4
V0	7	53	183	439	863
V1	497	383	563	79	973
V2	287	63	343	169	583
V3	627	343	773	959	943
V4	767	473	103	699	303

*Figura 5 - Representação dos Pesos entre Vértices*

Exemplo: Do V0 para o V0 têm um peso de 7, do V0 para V1 têm um peso de 53 e assim sucessivamente.



## Implementação

### Contar Ficheiro

```
/**
 * Esta funcao conta linhas do ficheiro.
 *
 * \param nomeFicheiro Variavel que contem o nome do ficheiro
 * \return
 */
int ContarFicheiro(char nomeFicheiro[]) {
    int c;
    FILE* fp = fopen(nomeFicheiro, "r");
    if (fp == NULL)
        return 0;

    //Conta linhas atraves de '\n'
    int linhas = 1;
    while ((c = fgetc(fp)) != EOF) {
        if (c == '\n')
            linhas = linhas + 1;
    }
    fclose(fp);
    return linhas;
}
```

Figura 6 - Função ContarFicheiro

Esta função tem como intuito de deixar o programa dinâmico. Através desta função que identifica quantas linhas tem o ficheiro, já não é necessário inserir um valor fixo ao número de vértices. Assim, à medida que o ficheiro cresce, o programa funciona sem necessidade de manutenção.

## Importação de Adjacências

```
/**
 * Esta funcao importa os valores do ficheiro.
 *
 * \param g Variavel que contem os valores do grafo
 * \param nomeFicheiro Variavel que contem o nome do ficheiro
 * \return
 */
int* ImportaValorFicheiro(Grafo* g, char nomeFicheiro[]) {
    FILE* fp = fopen(nomeFicheiro, "r");
    if (fp == NULL) return 0;

    //Aloca a memoria
    int* peso = (int*)malloc(sizeof(int) * (g->totalVertices * g->totalVertices));
    if (peso == NULL) return NULL;

    for (int i = 0; i < (g->totalVertices * g->totalVertices); i++)
    {
        //Guarda os valores do ficheiro
        fscanf(fp, "%d:", (peso + i));
    }

    fclose(fp);
    return peso;
}
```

Figura 7 - Função ImportaFicheiro

Esta função foi utilizada para importar os valores do ficheiro “Peso.txt”, que foram guardados num array dinâmico (peso + i) e posteriormente inseridos como peso para as adjacências.

## Análise de Resultados

### Operações não implementadas

Neste projeto não consegui concluir algumas funções, nomeadamente as que envolvem somas e calcular os caminhos (ligação entre vértices).

### Abordagens Exploradas

Não consegui carregar para o grafo as adjacências e o seu respetivo valor.

## Conclusão

Na minha perspetiva foi um trabalho que me ajudou a desenvolver intelectualmente, seja a nível de produção de código como em resolução de problemas, penso que está um trabalho bem conseguido, embora pudesse ter trabalhado mais. Acredito que tive uma prestação positiva.

## Trabalho a Desenvolver

Futuramente, pretendo concluir as funções que não conclui, e desenvolver um grafo que não seja quadrado, de forma a tornar o trabalho mais completo e a ganhar mais conhecimento sobre manipulação de grafos.

Trabalho realizado por:

Miguel Pires - 27968