

Relatório do Projeto de EDA - Segunda Fase: Implementação de Grafos para Simulação de Antenas

Bruno Paiva
a31496@alunos.ipca.pt

IPCA - Licenciatura de Sistemas Informáticos

Maio 2025

Resumo

Este relatório descreve o desenvolvimento da segunda fase do projeto da unidade curricular de Estruturas de Dados Avançadas, que consistiu na implementação de um sistema de simulação de efeitos de antenas utilizando grafos em linguagem C.

O trabalho aborda a modelação do problema, a estrutura de dados desenvolvida, as funcionalidades implementadas e os resultados obtidos.

A solução proposta permite carregar dados de antenas a partir de ficheiros, construir grafos de conexões entre antenas, realizar pesquisas em profundidade e largura, identificar caminhos entre antenas e detetar interseções entre conexões.

O projeto demonstra a aplicação prática de conceitos fundamentais de grafos e boas práticas de desenvolvimento de software.

Conteúdo

1	Introdução	2
1.1	Motivação	2
1.2	Objetivos	2
1.3	Metodologia	3
1.4	Estrutura do Documento	3
2	Descrição do Problema	4
2.1	Requisitos Funcionais	4
2.2	Requisitos Não Funcionais	5
3	Solução Implementada	6
3.1	Estrutura de Dados	6
3.1.1	Representação de Antenas	6
3.1.2	Representação de Arestas	6
3.1.3	Estrutura do Grafo	7
3.2	Arquitetura do Sistema	8
3.3	Principais Algoritmos	9
3.3.1	Construção do Grafo	9
3.3.2	Pesquisa em Profundidade (DFS)	9
3.3.3	Pesquisa em Largura (BFS)	9
3.3.4	Identificação de Caminhos	9
3.3.5	Detecção de Interseções	9
4	Exemplos de Funcionamento	10
5	Conclusões e Trabalho Futuro	11
6	Repositório e Documentação	12

Capítulo 1

Introdução

1.1 Motivação

O presente trabalho surge como continuação da primeira fase do projeto da unidade curricular de Estruturas de Dados Avançadas. Enquanto a primeira fase utilizou listas ligadas para resolver o problema proposto, esta segunda fase explora estruturas de grafos, consolidando conhecimentos mais avançados de teoria dos grafos e sua implementação em C.

1.2 Objetivos

Os principais objetivos desta fase foram:

- Desenvolver uma estrutura de dados em grafo para representar antenas e suas conexões
- Implementar algoritmos de pesquisa em grafos (DFS e BFS)
- Identificar caminhos entre pares de antenas
- Detetar interseções entre conexões de antenas
- Armazenar resultados em ficheiros binários
- Documentar o código seguindo boas práticas

1.3 Metodologia

A metodologia adotada seguiu as seguintes etapas:

1. Análise do problema e requisitos
2. Desenho da estrutura de dados em grafo
3. Implementação incremental das funcionalidades
4. Testes unitários e integração
5. Documentação do código
6. Redação do relatório técnico

1.4 Estrutura do Documento

O presente documento está organizado da seguinte forma: o Capítulo 2 descreve o problema em detalhe; o Capítulo 3 apresenta a solução implementada; o Capítulo 4 mostra exemplos de funcionamento; e finalmente o Capítulo 5 sumariza as conclusões e trabalhos futuros.

Capítulo 2

Descrição do Problema

O problema consiste em simular o comportamento de múltiplas antenas de transmissão representadas num grafo, onde:

- Cada vértice representa uma antena com posição (x,y) e frequência
- As arestas conectam antenas com a mesma frequência

2.1 Requisitos Funcionais

- Ler configurações de antenas a partir de ficheiro texto
- Construir grafo de conexões entre antenas
- Implementar pesquisa em profundidade (DFS)
- Implementar pesquisa em largura (BFS)
- Identificar todos os caminhos entre duas antenas
- Detetar interseções entre conexões
- Armazenar resultados em ficheiro binário

2.2 Requisitos Não Funcionais

- Implementação em linguagem C
- Uso obrigatório de estruturas de grafos e listas dinâmicas
- Modularização do código
- Documentação com Doxygen
- Validação de entrada de dados

Capítulo 3

Solução Implementada

3.1 Estrutura de Dados

A estrutura de dados principal foi implementada como um grafo não direcionado, conforme o seguinte código:

3.1.1 Representação de Antenas

Cada antena é modelada como um vértice contendo:

- Identificador único
- Coordenadas espaciais
- Frequência de operação
- Estado de visitação
- Lista de arestas adjacentes

3.1.2 Representação de Arestas

As ligações entre antenas são representadas por:

- Referência para a antena de destino
- Ponteiro para a próxima aresta

3.1.3 Estrutura do Grafo

O grafo principal contém:

- Contador do número total de vértices
- Ponteiro para a lista de vértices

3.2 Arquitetura do Sistema

O sistema foi organizado nos seguintes módulos principais:

- **main.c**: Contém a lógica principal do programa
- **estruturas.h**: Define as estruturas de dados
- **func.h**: Declaração das funções
- **func.c**: Implementação das funções
- **libfunc.a**: Biblioteca das funções
- **antenas.txt**: Ficheiro de entrada com a matriz de antenas
- **output.bin**: Ficheiro binário de saída
- **makefile**: Ficheiro para compilar e criar o executável

3.3 Principais Algoritmos

3.3.1 Construção do Grafo

O algoritmo de construção do grafo segue os seguintes passos:

1. Ler o ficheiro de entrada e contar o número de antenas
2. Alocar memória para o grafo
3. Inserir as antenas nos vértices do grafo
4. Criar arestas entre antenas com a mesma frequência

3.3.2 Pesquisa em Profundidade (DFS)

Implementação recursiva da DFS que visita todos os vértices alcançáveis a partir de um vértice inicial, registrando as arestas percorridas no ficheiro binário.

3.3.3 Pesquisa em Largura (BFS)

Implementação iterativa da BFS usando uma fila, que visita os vértices por níveis de distância do vértice inicial.

3.3.4 Identificação de Caminhos

Algoritmo recursivo que encontra todos os caminhos possíveis entre dois vértices, armazenando cada caminho encontrado no ficheiro binário.

3.3.5 Detecção de Interseções

Algoritmo que verifica se segmentos de reta entre pares de antenas se intersectam, utilizando cálculos de geometria.

Capítulo 4

Exemplos de Funcionamento

Para o seguinte mapa de entrada (antenas.txt):

```
.....  
.A..B.  
.....  
.B..A.  
.....  
.....
```

O programa produz a seguinte saída da leitura do ficheiro binário:

```
=== ANTENAS (4) ===  
Antena ID 3: (4, 3) - A  
Antena ID 2: (1, 3) - B  
Antena ID 1: (4, 1) - B  
Antena ID 0: (1, 1) - A  
=== DFS ===  
Antena DFS: (1, 3) - B  
Antena DFS: (4, 1) - B  
=== BFS ===  
Antena BFS: (4, 3) - A  
Antena BFS: (1, 1) - A  
=== CAMINHO ===  
Antena no caminho: ID 3  
Antena no caminho: ID 0  
=== INTERSEÇÕES ===  
Interseção: (1,1)-(4,3) com (4,1)-(1,3), freq: A  
Interseção: (4,1)-(1,3) com (1,1)-(4,3), freq: B
```

Capítulo 5

Conclusões e Trabalho Futuro

Através deste projeto foi possível consolidar conhecimentos fundamentais sobre estruturas de grafos e sua implementação em C. Os principais conhecimentos desenvolvidos foram:

- Domínio da manipulação de grafos em C
- Implementação de algoritmos de pesquisa em grafos
- Técnicas de armazenamento em ficheiros binários
- Importância da modularização e documentação de código

Como trabalho futuro, sugere-se:

- Implementação de algoritmos mais eficientes para detecção de interseções
- Visualização gráfica do grafo e dos caminhos encontrados

Capítulo 6

Repositório e Documentação

Todo o código está disponível no repositório GitHub: https://github.com/yesisr/TP_EDA_Fase-2

A documentação do código foi gerada com Doxygen e está incluída no repositório na pasta docs/doxygen.