

Instituto Politécnico do Cávado e do Ave
Escola Superior de Tecnologia

Trabalho Prático Estruturas de Dados Avançadas
Fase 2

João Pedro Rocha Novais
a24825

Índice

Conteúdo

Índice.....	1
Índice de figuras	2
Resumo.....	2
1. Introdução	3
1.1 Motivação	3
1.2 Enquadramento.....	3
1.3 Objetivos	3
1.4 Metodologia de Trabalho	4
1. Análise e compreensão do problema:.....	4
2. Planeamento da arquitetura do sistema:	4
3. Implementação incremental:.....	4
4. Testes e validação:	4
5. Revisão e otimização:.....	4
6. Documentação:.....	5
2. Estrutura do Documento	5
3. Estrutura de dados.....	6
4 Funcionalidades implementadas	8
4.1 Manipulação de Antenas	8
4.2 Conectividade entre Antenas.....	9

4.3 Cálculo de Efeitos Nefastos	9
4.4 Algoritmos de Pesquisa e Travessia	9
4.5 Ficheiros	10
5. Dificuldades encontradas ao longo do projeto	10
6. Conclusões	10
7. Anexos	11
7.1 Código-fonte	11

Índice de figuras

Figura 1 Struct VAntena.....	6
Figura 2 Struct Adjacente	7
Figura 3 Struct Grafo	7
Figura 4 Struct EfeitoNefasto	8

Resumo

Este relatório apresenta o desenvolvimento de um Sistema de Gestão de Antenas e Efeitos Nefastos, concebido no âmbito da Unidade Curricular de Estruturas de Dados Avançadas. O sistema implementa uma estrutura de dados baseada em grafos e listas ligadas na linguagem C, permitindo a representação e manipulação de antenas de diferentes frequências distribuídas numa matriz bidimensional. O projeto foca-se na deteção de configurações específicas que geram efeitos nefastos, aplicando algoritmos de pesquisa e travessia em grafos. Foram desenvolvidas funcionalidades de manipulação de antenas, estabelecimento de conectividade, cálculo de efeitos nefastos e persistência de dados. As principais dificuldades encontradas relacionaram-se com a implementação de algoritmos de deteção de interseções e com a gestão eficiente da memória em estruturas dinâmicas.

1. Introdução

Este projeto foi desenvolvido como trabalho prático para a Unidade Curricular de Estruturas de Dados Avançadas (EDA), integrada no 2º semestre do 1º ano do curso. O objetivo principal foi aplicar os conhecimentos adquiridos sobre estruturas de dados dinâmicas na linguagem C, particularmente a implementação de grafos e listas ligadas.

O sistema desenvolvido permite a gestão de antenas de diferentes frequências distribuídas numa matriz bidimensional, calculando os efeitos nefastos causados por configurações específicas dessas antenas.

Este documento apresenta as estruturas de dados utilizadas, as funcionalidades implementadas, os algoritmos aplicados e as dificuldades encontradas durante o desenvolvimento.

1.1 Motivação

Este trabalho surge da necessidade de desenvolver ferramentas computacionais capazes de modelar estes sistemas complexos, utilizando estruturas de dados dinâmicas que permitam representar eficientemente as relações espaciais entre antenas de diferentes frequências e identificar automaticamente pontos críticos (nefastos) resultantes da interação entre estas.

1.2 Enquadramento

Este projeto foi desenvolvido no âmbito da Unidade Curricular de Estruturas de Dados Avançadas, integrada no 2º semestre do 1º ano do curso de Licenciatura em Engenharia de Sistemas Informáticos do Instituto Politécnico do Cávado e do Ave. O trabalho visa consolidar os conhecimentos adquiridos ao longo do semestre, com especial ênfase na definição e manipulação de estruturas de dados dinâmicas na linguagem de programação C.

1.3 Objetivos

Os objetivos principais deste projeto foram:

1. **Desenvolver um sistema de gestão de antenas** capaz de representar, manipular e analisar configurações de antenas de diferentes frequências numa matriz bidimensional.
2. **Implementar estruturas de dados adequadas**, nomeadamente grafos e listas ligadas, para modelar eficientemente as relações espaciais e as conexões entre antenas.
3. **Criar algoritmos de deteção de efeitos nefastos** que identifiquem padrões geométricos específicos formados por antenas de mesma frequência, que possam resultar em interferências ou outros efeitos indesejados.

4. **Aplicar algoritmos de pesquisa e travessia de grafos** para explorar a estrutura de conexões entre antenas e identificar características relevantes da rede.
5. **Guardar dados** garantir que os dados são guardados em ficheiros .bin.

1.4 Metodologia de Trabalho

A metodologia adotada para o desenvolvimento deste projeto seguiu uma abordagem estruturada e iterativa, dividida nas seguintes fases:

1. Análise e compreensão do problema:

- Estudo aprofundado dos requisitos do projeto
- Definição clara do conceito de "efeito nefasto" e dos critérios para a sua deteção
- Identificação das operações necessárias para a manipulação de antenas e suas conexões

2. Planeamento da arquitetura do sistema:

- Definição das estruturas de dados principais (Antena, Adjacente, Grafo, EfeitoNefasto)
- Estabelecimento das relações entre os diferentes componentes do sistema

3. Implementação incremental:

- Desenvolvimento inicial das estruturas de dados básicas
- Implementação das operações fundamentais de manipulação de antenas
- Desenvolvimento dos algoritmos de conexão entre antenas
- Implementação dos algoritmos de deteção de efeitos nefastos
- Desenvolvimento das funcionalidades de persistência de dados

4. Testes e validação:

- Criação de casos de teste para verificar o comportamento do sistema
- Validação das funcionalidades implementadas com diferentes configurações de antenas
- Verificação da correta deteção de efeitos nefastos
- Testes de desempenho para avaliar a eficiência dos algoritmos

5. Revisão e otimização:

- Análise crítica do código implementado

- Identificação e resolução de bugs e inconsistências
- Otimização de algoritmos para melhorar o desempenho

6.Documentação:

- Elaboração do relatório técnico detalhando todas as componentes do sistema
- Documentação do código-fonte com comentários explicativos
- Criação de exemplos de utilização e casos de teste

Esta metodologia permitiu um desenvolvimento progressivo e controlado do sistema, facilitando a identificação e resolução de problemas em fases iniciais e garantindo a qualidade do produto final.

2. Estrutura do Documento

Este relatório técnico está organizado de forma a apresentar de maneira clara e estruturada todos os

aspectos relevantes do projeto desenvolvido:

1. Introdução: Apresenta o contexto geral do projeto, o seu âmbito e os seus objetivos fundamentais.
2. Motivação: Explica as razões que levaram ao desenvolvimento do sistema, tanto do ponto de vista prático como académico.
3. Enquadramento: Contextualiza o projeto no âmbito mais amplo da gestão de infraestruturas de telecomunicações e da aplicação de estruturas de dados avançadas.
4. Objetivos: Detalha os objetivos específicos que o projeto se propôs alcançar.
5. Metodologia de Trabalho: Descreve a abordagem utilizada para o desenvolvimento do projeto, incluindo as fases de trabalho e as práticas adotadas.
6. Estrutura de dados: Apresenta em detalhe as estruturas de dados implementadas, incluindo a sua definição, atributos e relações.
7. Funcionalidades implementadas: Descreve as várias funcionalidades do sistema, organizadas por categorias:


- Manipulação de Antenas
- Conectividade entre Antenas
- Cálculo de Efeitos Nefastos
- Algoritmos de Pesquisa e Travessia
- Ficheiro

3. Estrutura de dados

Para este projeto, foram implementadas quatro estruturas de dados principais:

Struct VAntena

Esta estrutura representa uma antena no sistema, com:



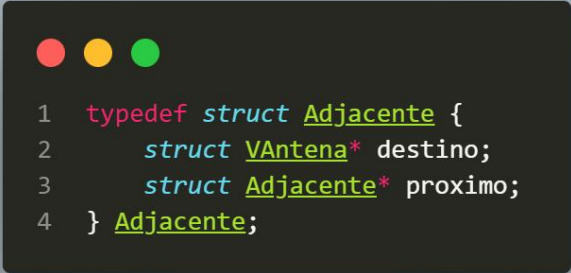
```
1 typedef struct VAntena {  
2     char frequencia;  
3     int linha, coluna;  
4     bool visitado;  
5     struct Adjacente* listaAdj;  
6     struct VAntena* proximo;  
7 } VAntena;
```

Figura 1 Struct VAntena

- frequencia : Identificador da frequência da antena (carácter)
- linha, coluna : Coordenadas da antena na matriz
- visitado : Flag utilizada nos algoritmos de pesquisa
- listaAdj : Lista de antenas adjacentes (conexões)
- proximo : Apontador para a próxima antena na lista principal

Struct Adjacente

Esta estrutura representa uma conexão entre antenas:



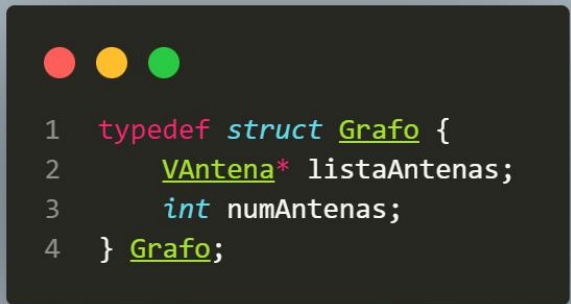
```
1 typedef struct Adjacente {  
2     struct VAntena* destino;  
3     struct Adjacente* proximo;  
4 } Adjacente;
```

Figura 2 Struct Adjacente

- destino : Apontador para a antena de destino
- proximo : Apontador para a próxima conexão na lista de adjacências

Struct Grafo

Estrutura principal que agrega todas as antenas:



```
1 typedef struct Grafo {  
2     VAntena* listaAntenas;  
3     int numAntenas;  
4 } Grafo;
```

Figura 3 Struct Grafo

- listaAntenas : Lista ligada de todas as antenas
- numAntenas : Contador do número total de antenas

Struct EfeitoNefasto

Estrutura que representa um ponto com efeito nefasto:

A screenshot of a code editor with a dark background and three colored window control buttons (red, yellow, green) in the top-left corner. The code is written in C and defines a struct named EfeitoNefasto. The code is as follows:

```
1 typedef struct EfeitoNefasto {  
2     int linha, coluna;  
3     char frequencia;  
4     struct EfeitoNefasto* proximo;  
5 } EfeitoNefasto;
```

Figura 4 Struct EfeitoNefasto

- linha, coluna : Coordenadas do ponto com efeito nefasto
- frequencia : Identificador da frequência que causou o efeito
- proximo : Apontador para o próximo efeito na lista

4 Funcionalidades implementadas

4.1 Manipulação de Antenas

O sistema permite operações básicas de gestão de antenas, como:

- **Criação de antenas:** Através da função criarAntena() , é possível criar uma nova antena com uma frequência e posição específicas.
- **Inserção de antenas:** Com a função inserirAntena() , novas antenas são adicionadas ao grafo, verificando previamente se a posição está livre.
- **Remoção de antenas:** A função RemoverAntena() permite eliminar uma antena do grafo, enquanto removerVertice() também remove todas as conexões associadas.
- **Verificação de posição:** A função posicaoOcupada() verifica se já existe uma antena em determinadas coordenadas.

Estas funcionalidades são fundamentais para a construção e manutenção da estrutura do grafo

4.2 Conectividade entre Antenas

Para estabelecer conexões entre antenas:

- **Conexão de antenas:** A função `conectarAntenasMesmoTipo()` cria automaticamente ligações entre antenas de mesma frequência, gerando um grafo não direcionado.
- **Remoção de conexões:** A função `removerAdjacente()` permite eliminar conexões específicas entre duas antenas.

Estas funções são essenciais para modelar as interações entre antenas de mesma frequência.

4.3 Cálculo de Efeitos Nefastos

Uma das principais funcionalidades do sistema é calcular os efeitos nefastos:

- **Deteção de efeitos nefastos:** A função `calcularEfeitosNefastos()` identifica pontos específicos onde duas antenas de mesma frequência causam um efeito nefasto.
- **Gestão de efeitos:** As funções `CriarNafsto()`, `inserirEfeito()` e `existeEfeito()` permitem criar, inserir e verificar a existência de efeitos nefastos na lista.
- **Visualização de resultados:** A função `imprimirEfeitosNefastos()` amostra todos os efeitos nefastos detetados.

4.4 Algoritmos de Pesquisa e Travessia

Para explorar o grafo, foram implementados algoritmos de pesquisa:

- **Pesquisa em profundidade (DFS):** A função DFS() implementa o algoritmo de pesquisa em profundidade para percorrer o grafo.
- **Gestão de visitados:** A função limparVisitados() reinicia as variáveis de visitado em todas as antenas.

Estes algoritmos demonstram a utilidade das estruturas de dados para exploração e análise do grafo.

4.5 Ficheiros

5. Dificuldades encontradas ao longo do projeto

1. Implementação do carregamento de dados: A leitura de ficheiros e conversão para a estrutura do grafo exigiu atenção aos detalhes para evitar erros de alocação de memória.
2. Algoritmos de deteção de interseções: A implementação do algoritmo para detetar quando Antenas se cruzam.
3. Gestão de memória: O trabalho com estruturas dinâmicas exigiu atenção constante para evitar fugas de memória, especialmente ao remover elementos do grafo.

Para superar estes desafios, foi necessário estudar exemplos de código fornecidos nas aulas, realizar pesquisas adicionais e discutir abordagens com colegas.

6. Conclusões

A implementação do sistema permitiu consolidar diversos conceitos teóricos fundamentais, nomeadamente a representação de grafos em memória, os algoritmos de travessia e pesquisa, e a gestão eficiente de estruturas de dados dinâmicas. Além disso, o projeto exigiu e fomentou o desenvolvimento de competências essenciais em programação, como a gestão criteriosa de memória, a implementação de algoritmos complexos e a criação de mecanismos de persistência de dados.

Os objetivos inicialmente estabelecidos foram alcançados com sucesso, resultando num sistema funcional capaz de:

- Representar antenas de diferentes frequências numa matriz bidimensional
- Estabelecer e gerir conexões entre antenas de mesma frequência

- Detetar e registar efeitos nefastos baseados em padrões geométricos específicos
- Aplicar algoritmos de pesquisa para explorar a estrutura do grafo
- Assegurar a persistência dos dados através de operações de carregamento e armazenamento

As dificuldades encontradas durante o desenvolvimento, particularmente na implementação de algoritmos de deteção de interseções e na gestão de memória, proporcionaram importantes oportunidades de aprendizagem e aprofundamento dos conhecimentos. A superação destes desafios contribuiu significativamente para o crescimento técnico e para o desenvolvimento de estratégias eficazes de resolução de problemas.

7. Anexos

7.1 Código-fonte

Repositório Git: https://github.com/Joao-pnovais/EDA_Fase2