



UNIVERSIDADE
DE ÉVORA



Departamento de Informática - Universidade de Évora

Trabalho de EDA2 2023/24

“ Assignment 2: Lost in the Enchanted Forest ”

Diogo Matos Aluno 54466

Ivo Rego Aluno 32640

30 de abril de 2024

1 Introdução

- O presente relatório detalha a implementação de um programa que resolve o desafio de "**Retorno à Origem**" inspirado no conto clássico de Hansel e Gretel.

No contexto do problema, Hansel e Gretel, após serem deixados na floresta pela madrasta, utilizam cristais mágicos para marcar seu caminho em busca de uma rota de volta para casa, evitando os perigos e desafios da floresta encantada.

O objetivo deste trabalho é utilizar conceitos avançados de estruturas de dados e algoritmos para modelar e resolver o problema de encontrar o caminho mais curto de volta ao ponto de partida após uma série de movimentos dentro de uma grid bidimensional.

Este problema é abordado através da construção de um grafo, onde cada vértice representa um ponto específico no caminho percorrido por Hansel e Gretel, e cada aresta representa um possível movimento entre dois pontos. A solução requer o cálculo do caminho mínimo utilizando uma pesquisa em largura (BFS), partindo do último ponto alcançado de volta ao ponto inicial.

Este problema não só testa a habilidade de implementação de estruturas de dados complexas, como grafos e filas, mas também a capacidade de aplicar algoritmos de pesquisa para resolver problemas de caminhos mínimos utilizando grafos.

2 Metodologia

- A metodologia utilizada neste projeto envolve várias etapas cruciais para o desenvolvimento e análise do algoritmo, detalhadas abaixo:

Especificação do Problema: Compreender em profundidade o problema apresentado, incluindo suas regras, limitações e o contexto de aplicação. Para isso, foi realizada uma revisão da narrativa de Hansel e Gretel adaptada ao contexto dos cristais mágicos.

Levantamento de Requisitos: Identificar os requisitos funcionais e não funcionais do sistema, incluindo o processamento dos movimentos, gerar o grafo e a implementação do algoritmo de pesquisa para encontrar o caminho de retorno.

Desenho da Solução: Escolher as estruturas de dados adequadas (grafos, listas, mapas) para representar o problema. Definir o algoritmo de pesquisa em largura (BFS) como método de resolução para o cálculo do caminho mínimo de volta ao ponto de partida.

Implementação: Implementar a solução em Java, fazendo uso intensivo de estruturas de dados como HashMap para mapeamento de vértices e Queue para a gestão dos vértices durante a BFS.

Testes: Foram realizados testes para assegurar que todos os componentes funcionam conforme esperado e que o algoritmo retorna o número correto de movimentos mínimos.

Otimização: O código foi analisado e otimizado para melhorar a eficiência, especialmente considerando a complexidade temporal e espacial.

Documentação e Relatório: Documentação de todo o processo de desenvolvimento e dos resultados no presente relatório, que inclui a descrição do algoritmo, exemplos ilustrativos e análise de complexidade.

3 Descrição do Algoritmo

- O programa implementado tem como objetivo calcular o número mínimo de movimentos necessários para que Hansel e Gretel retornem ao ponto de partida após uma jornada através de uma floresta encantada, marcando o seu caminho com cristais mágicos. **O algoritmo realiza as seguintes etapas principais:**

Leitura e Armazenamento dos Movimentos: Captura e armazena os movimentos realizados pelos personagens a partir de entradas padrão. Os movimentos são lidos do input como uma sequência de caracteres ('N', 'E', 'S', 'W') e armazenados numa String.

Construção e Mapeamento do Grafo: Cada movimento gera um vértice em um grafo. Um grafo é construído onde cada vértice representa uma posição no caminho percorrido, incluindo coordenadas (x, y) e as adjacências deste, ou seja, o seu vértice filho e vértice pai.

Mapeamento dos Vértices: Os vértices são mapeados num HashMap baseado nas suas coordenadas para rápida recuperação e verificação de existência durante a pesquisa no grafo. Este mesmo HashMap, para cada key (posição) guardada, tem uma lista de vértices pois pode haver mais do que um vértice com a mesma coordenada.

Pesquisa em Largura (BFS): Uma pesquisa em largura é iniciada a partir do último vértice (a casa da bruxa) para encontrar o caminho mais curto de volta ao ponto de partida, explorando cada vértice e verificando adjacências conforme definido pelos possíveis movimentos.

Cálculo do Caminho de Retorno: A distância de cada vértice ao ponto de partida é atualizada durante a BFS. O comprimento do caminho mais curto é determinado pela distância do último vértice processado até à casa da bruxa.

4 Descrição do Grafo construído

- Os vértices do grafo representam posições específicas alcançadas pelos personagens, com arestas indicando os possíveis movimentos. Cada vértice mantém informações sobre a sua posição, o movimento que o originou e suas conexões com outros vértices (adjacências). O grafo tem vértices representando cada passo do caminho. Cada vértice tem um identificador baseado na sua coordenada e está ligado aos seus adjacentes diretos conforme os movimentos de Hansel e Gretel. A construção do grafo permite a verificação de caminhos cruzados e o cálculo do caminho de retorno mais eficiente (mais curto).

Para o seguinte input:

2

ENNWNNESSNNEE

SSWSSE

(0,0) -> (1,0) -> (1,1) -> (1,2) -> (0,2) -> (0,3) -> (0,4) -> (1,4) -> (1,3) -> (1,2) -> (1,3) -> (1,4) -> (2,4) -> (3,4) -> (3,3) -> (3,2) -> (2,2) -> (2,1) -> (2,0) -> (3,0)

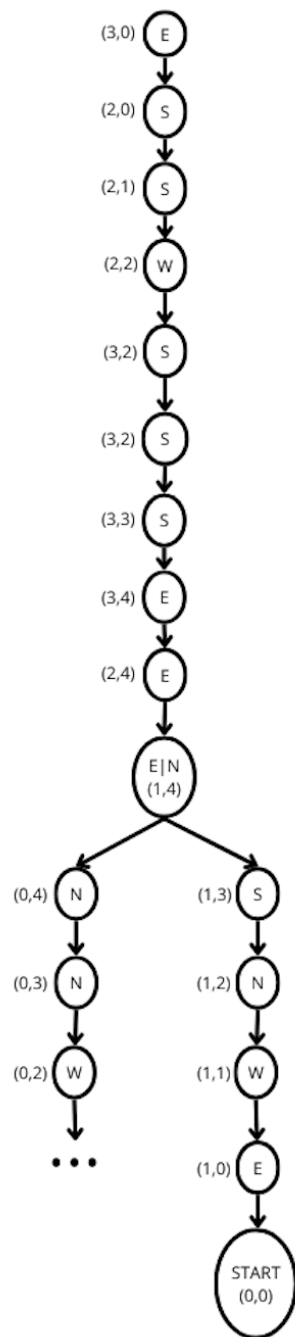


Figura 1: Representação visual do grafo

5 Cálculo de complexidades

- **Complexidade Temporal:** O programa realiza a BFS em um grafo onde cada vértice é visitado no máximo uma vez, e cada aresta é explorada uma vez. Assim, a complexidade temporal no pior dos casos é

$$O(L) + O(V) + O(2) + O(V)O(4)O(V)O(2) = O(L) + O(V) + O(V^2)$$

onde V é o número de vértices possíveis, sendo o primeiro a leitura e armazenamento dos vértices e o segundo a pesquisa BFS no pior dos casos. L é o número de linhas de movimentos a ler do input.

- **Complexidade Espacial:** A memória é principalmente utilizada para armazenar os vértices e o mapeamento das coordenadas. Cada movimento gera um novo vértice e possíveis adjacências.

A complexidade espacial no pior dos casos é

$$O(N) + O(N^2)$$

sendo que N o número de vértices possíveis, onde o primeiro representa o array vertices e o segundo representa o HashMap, onde para N coordenadas pode ter-se N vértices.

6 Conclusão e Comentários Adicionais

- O algoritmo implementado oferece uma solução robusta e eficaz para o problema de encontrar o caminho de retorno para Hansel e Gretel. A metodologia aplicada garantiu uma abordagem sistemática e estruturada, facilitando a análise e otimização do programa. Este relatório e a metodologia usada ilustram não só a solução de um problema complexo através de estruturas de dados e algoritmos mas também destacam a importância de um processo de desenvolvimento bem definido e documentado no contexto da programação.