Fifteen-Puzzle Solver

Duc Nguyen

February 18, 2025

1 Introduction

This repository contains a C++ solution for the **Fifteen-Puzzle** problem, an extension of the classic eight-puzzle. In this project, two search algorithms are implemented — one uninformed and one informed — to solve the puzzle and find the sequence of moves leading to the goal state.

2 Project Description

The Fifteen-Puzzle is a sliding puzzle that consists of a 4x4 grid of tiles with one empty space. The tiles are labeled with the letters A through O, and the empty space is represented by a hyphen (`-`). The goal is to rearrange the tiles by sliding them into the empty space until the puzzle reaches the solution:

```
A B C D
E F G H
I J K L
M N O -
```

2.1 How It Works

- **Puzzle Mechanics:** Only tiles adjacent to the empty space can be slid into that empty spot. This is often easier to conceptualize as moving the empty space rather than the tile.
- **Search Algorithms:**
  - **Uninformed Search:** Implements Breadth-First Search (BFS) to explore the state space without using any heuristic information.
  - **Informed Search:** Implements an algorithm like A* to efficiently guide the search towards the goal state.
- **Input/Output:** The program reads puzzle states from text files located in a dedicated folder (e.g., `puzzle_files`). Each file contains four rows of four characters, separated by spaces. The program then prints the initial state, allows the user to select the search algorithm, and finally outputs:
  - The complete path from the starting state to the goal.

- The number of moves required (path length).
- The total number of nodes expanded during the search.

3 Folder Structure

```
fifteen-puzzle-solver/
 WelcomeToMySolver.pdf   % Guide for the Solver
 main.cpp            % Source codes
 tested_files/       % tested_files that your inputs
```

4 Getting Started

4.1 Prerequisites

- A C++ compiler supporting C++11 (or later)
- Basic knowledge of command-line operations

4.2 How to Build and Run

1. **Clone the Repository:**

   ```
   git clone https://github.com/yourusername/fifteen-puzzle-solver.git
   cd fifteen-puzzle-solver
   ```

2. **Place Your Puzzle Files:**
   Save your puzzle input files (plain text files with a 4x4 grid) in the `puzzle_files` folder. For example, a file representing the goal state should look like:

   ```
   A B C D
   E F G H
   I J K L
   M N O -
   ```

3. **Compile the Code:**

   ```
   g++ -std=c++11 main.cpp -o fifteen_puzzle_solver
   ```

4. **Run the Executable:**

   ```
   ./fifteen_puzzle_solver
   ```

5. **Follow On-Screen Prompts:**
   Enter the file name (e.g., `test2.txt`) when prompted, choose the search algorithm, and the program will display the solution path, number of moves, and nodes expanded.

5 Acknowledgements

This project was developed as part of CSC 330's AI Project 2 assignment. Special thanks to the course instructors for providing the guidelines and problem statement.