

Problem Solving: Sliding Puzzle

In this task, your problem solving skills are required. This means that you have to be more creative rather than having a deep knowledge of a specific programming language or usage of certain powerful libraries. There are different ways on how to solve the problem at hand. Your solutions should be as elegant and efficient as possible.

Graph traversal and search algorithm can not only be used to find the shortest path in a maze or map, but also to schedule actions or to devise plans. The “map”, or the state-space, is built up over time and the graph edges are the possible action.

The objective is to use the A* algorithm to implement a solver for a simple sliding puzzle, in order to create a plan on how to move the moving puzzle pieces (see figure 1).

Note: Make sure to use the proper algorithm and data structures! With the right implementation, the calculations for the 3×3 puzzle should not take more than a few seconds. You can use appropriate standard libraries; no need to re-invent the wheel!

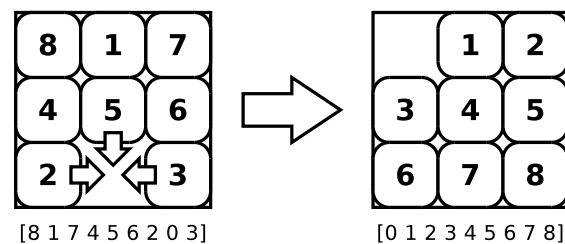


Figure 1: The start and the desired target state of the puzzle; the small arrows indicate the possible actions.

Tasks

1. Implement the A*-algorithm for this puzzle and calculate the shortest path from the start state [8 1 7 4 5 6 2 0 3] to the target state [0 1 2 3 4 5 6 7 8] (see Figure 1). Come up with a suitable heuristic for the quality of the current state. How long is the shortest path and what actions is it made up of?
2. How many solvable states are there, i.e. states from which the goal-state can be reached? And how long is the longest path from any unordered but solvable state to the ordered goal-state? (Hint: You might want to use a different search algorithm for this.)
3. (optional) Implement the algorithm in such a way that it can also be used for sliding puzzles with different dimensions, or for entirely different problems.

Besides the actual source code, please include a description how to start and run the code, as well as a description of how your program works, and why you decided to use this approach.