Jenna Flannery

September 18, 2024

Artificial Intelligence

Project 1 Report

Introduction

This program uses breadth first search and A* search to solve the 8-puzzle problem. It randomly generates three initial states and implements both algorithms on each. Because not every initial state can be solved, each algorithm will time out after 10 seconds of searching in order to move on to the next initial state.

Results and Analysis

Both algorithms behave correctly. Technically, with this implementation, A* search generates nodes slightly slower than breadth first search because calculating the mismatched tiles and checking the closed list for each child node has a run time of $O(n^3)$ compared to breadth first search's $O(n^2)$. However, popping from A*'s heap (which represents a priority queue) takes less time than from breadth first search's queue. Additionally, in many cases, A* search does not have to generate as many nodes to find the goal state as breadth first search. This is because of its heuristic function. Therefore, when running the program using seed 1054, A* search was able to find the goal state in more cases than breadth first search. On the first run, breadth first search generates about 60,000 nodes, but times out before it can find the goal. However, A* search only generates about 10,000 nodes and finds the goal at depth 21. I chose seed 1054 because it shows the efficiency that a heuristic function can add to a search algorithm.