CSC 480 Assignment 1 presentation:

Hi, my name is Minh

This is my CSC 480 Assignment 1 demonstration: We’re supposed to implement various search algorithms to solve the 8-puzzle.

First Let us demonstrate how one might run this program.

Even though I’m on Windows. This program was written in cross-platform C++11, so you can pull the code to test on Mac or Linux.

1. Describe/show how you modeled the 8-puzzle including the successor function.

So basically we have a State class.

To model the successor function, I implemented a function that generates possible states. The maximum number of possible states is always 4.

1. Describe/show how you implemented the search algorithms

Let us talk about Uninformed search: BFS, DFS, and UCS.

For BFS: I implemented with a FIFO queue data structure. For each state that’s expanded i.e popped off the queue, I marked it as visited and generate other possible states.

In DFS: We do the same except that we use a FILO stack data structure.

In UCS: We technically replace the normal queue in BFS with a priority queue, where the specified cost of each move equals the moving tile.

Next, consider informed search algorithms: GBFS and A\*.

In GBFS: We also use priority queues here. But we also check for for repeated states, both expanded and those that are still on the queues.

A\* has two variations:

1\* using h(n) = same heuristic as GBFS.

2\* using h(n) = the sum of the Manhattan distance between the current state and the goal state.

Describe/show how you checked for duplicate states.

I check for expanded and those that are still on the queues using a unique identification called state IDs. It’s basically a string representation of a state.

In Informed Search Algorithms, these state IDs are stored in a hash table, mapping to its lowest heuristic.

I have seen that only DFS failed in my implementation even though I did check for repeated states.