Henry Jacobs

CS 380 – Artificial Intelligence

May 12th , 2022

CS 380 – Artificial Intelligence Documentation

My program does contain the applyRule, shuffle, goal, and A\* hueristic functions and does not contain the Breadth-First, Depth-First, or Best-First functions. I tried for many nights to write the functions and struggled to understand how to implement the necessary data-structures in order to represent the states of the Rubik’s Cube. I’ve had a bit of a mental block this quarter (not asking for sympathy) and had a really hard time wrapping my head around this assignment.

The A\* algorithm I wrote will compare a goal string and a given state string and will check to see if each piece is in the right place. For the pieces not in the right place, it will calculate the distance from the piece’s current location to where it should be in the string. The dictionary starting positions is used to reference where the starting positions to where each letter should start in the goal string. A second dictionary (starting\_positions2) is updated every time a letter is found to be in the wrong place. This was done to reflect that a letter has taken that position and the next time the same letter is found to be in the wrong place, it will have to go in front of the one that has already been moved to be in the right position. I then sum all of the distances to the goal string in the variable return\_counter. The larger the number, the further it is from the goal. Nodes with the smallest distance should be chosen first. I used absolute value so that it’s easy to sort and understand that nodes with the smallest distance are closest to the goal state.

I have included output that shows the a\*algorithm running properly by testing it on various states.