Part 2 (Bonus): 8-Puzzle

Aravind Sundaresan (sundrsn2)

Krish Masand (masand2)

Rodney Shaghoulian (shaghou2)

(3 unit group)

The 8-puzzle is a very popular puzzle problem that can be solved with the A* Algorithm. The 8-puzzle is implemented using an array of ints of size 9. The algorithm uses 3 heuristics:

1. The number of misplaced tiles. This is calculated by passing in the list array to the heuristic method and checking which tiles aren't in the goal position (a second array of equal size used to represent the goal state)

2. Sum of the Manhattan distances between every tile's current position and desired position. Similar to step one, this is done by comparing all 8 of the tiles to the 8 tiles in their goal position in the solution list and summing the distances.

3. A value called Gaschnig's heuristic. Simply stated, this heuristic is the number of steps it would take to solve the problem if it was possible to swap any tile with the "space." It is admissible since it underestimates the distance for the solution of the problem with a closer approximation. It is implemented using two lists. One represents the current permutation, and the other is the location of the element i in the permutation array.

The algorithm is implemented using two lists, one open and ready to poll, and the other with new states expanded and ready to be processed. At the beginning of the algorithm, the initial state is put in the current permutation list. During each step, the best state according to our heuristics is removes from the state list. If the state is the goal state, the algorithm terminates. Otherwise, the state is expanded and the best state from the possible states is places on the second list. States from the second list move to the front list based on two conditions:

The successor hasn't already been added to either list, or it's on the list with a lower cost.