Zachary McNulty (zmcnulty, 1636402)

CSE 415 HW3: Heuristics and A\* Search

**Item 8: Heuristics for the Eight Puzzle**

**Uniform Cost Search (No Heuristics)**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Problem** | **Puzzle** **Permutation** | **Success** | **Aborted** | **Solution Length** | **Total Cost** | **States Expanded** | **Max OPEN Length** |
| A | [3,0,1,6,4,2,7,8,5] | Y | N | 7 | 7 | 166 | 101 |
| B | [3,1,2,6,8,7,5,4,0] | Y | N | 12 | 12 | 1490 | 898 |
| C | [4,5,0,1,2,8,3,7,6] | Y | N | 14 | 14 | 4070 | 2290 |
| D | [0,8,2,1,7,4,3,6,5] | Y | N | 16 | 16 | 7982 | 4700 |

**A\* Search With Hamming Heuristic**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Problem** | **Puzzle** **Permutation** | **Success** | **Aborted** | **Solution Length** | **Total Cost** | **States Expanded** | **Max OPEN Length** |
| A | [3,0,1,6,4,2,7,8,5] | Y | N | 7 | 7 | 7 | 6 |
| B | [3,1,2,6,8,7,5,4,0] | Y | N | 12 | 12 | 94 | 72 |
| C | [4,5,0,1,2,8,3,7,6] | Y | N | 14 | 14 | 189 | 127 |
| D | [0,8,2,1,7,4,3,6,5] | Y | N | 16 | 16 | 589 | 368 |

**A\* Search With Manhattan Heuristic**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Problem** | **Puzzle** **Permutation** | **Success** | **Aborted** | **Solution Length** | **Total Cost** | **States Expanded** | **Max OPEN Length** |
| A | [3,0,1,6,4,2,7,8,5] | Y | N | 7 | 7 | 7 | 6 |
| B | [3,1,2,6,8,7,5,4,0] | Y | N | 12 | 12 | 33 | 25 |
| C | [4,5,0,1,2,8,3,7,6] | Y | N | 14 | 14 | 56 | 39 |
| D | [0,8,2,1,7,4,3,6,5] | Y | N | 16 | 16 | 148 | 96 |

* All heuristic runs completed within a second or two

**Evaluating my Custom Heuristic**

My heuristic works by counting the total distance each face is from its desired face. For example, if a color is on the right face, its distance is zero. If it is on the completely opposite face, its distance is two. Else, its distance is one. This is pretty much just an extension to the Manhattan Heuristic to the Rubik2Cube problem. Below are some statistics surrounding its performance vs Hamming and UCS. In each trial, the solved cube was mixed up by 5 random moves before the program attempts to solve it. For consistency, the random moves were seeded by the same number for testing each heuristic so they all began from the same initial state.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Heuristic Type** | **Success?** | **Solution Length** | **Total Cost** | **State Expanded** | **Runtime** | **Max OPEN length** |
| UCS | No | N/A | N/A | N/A | >5 minutes | N/A |
| Hamming | Yes | 7 | 7 | 105 | 1.71 sec | 1156 |
| Custom | Yes | 5 | 5 | 60 | 0.59 sec | 661 |

Note that Hamming did NOT produce an optimal solution. This is because for this problem, 8 faces move at a time. Thus, counting the number of faces out of place is not an admissible heuristic: a cube one move away from being solved has 8 faces out of place. Even though my custom heuristic is also not admissible (consider the same case where the cube is one move away from being solved) it appears to greatly reduce the number

of states required to solve the cube as well as the length of the OPEN list. I reran these tests several times to see if this holds up on different starting positions and this seems to be the case. Comparing the costs of the two heuristics, it is clear the Hamming heuristic is O(n^2), where n is the side length of the cube: you have to check the color of 6\*n^2 faces. My custom heuristic is O(n^2) as it also only traverses the set of faces a single time. So it gets improved performance at no extra computational cost. Of course, it is still a significant cost.