2001CS05, 2001CS23, 2001CS29

CS571 Assignment 2

A star search

Program output for various examples:

```
Enter the input matrix
1 2 3
4 5 6
7 0 8
For h1:
PASS
1 2 3
4 5 6
7 0 8

1 2 3
4 5 6
0 7 8

1 2 3
0 5 6
4 7 8

1 2 3
5 0 6
4 7 8

1 2 3
5 6 0
4 7 8

1 2 3
5 6 8
4 7 0

1 2 3
5 6 8
4 0 7

1 2 3
5 0 8
4 6 7
```

```
1 2 3
0 5 8
4 6 7

1 2 3
4 5 8
0 6 7

1 2 3
4 5 8
6 0 7

1 2 3
4 5 8
6 7 0

1 2 3
4 5 0
6 7 8

1 2 3
4 0 5
6 7 8

States on optimal path: 14
Time taken (ms): 138.662
The total number of states explored: 2297
For h2:
PASS
1 2 3
4 5 6
7 0 8

1 2 3
4 5 6
0 7 8

1 2 3
0 5 6
4 7 8

1 2 3
5 0 6
```

```
4 7 8

1 2 3
5 6 0
4 7 8

1 2 3
5 6 8
4 7 0

1 2 3
5 6 8
4 0 7

1 2 3
5 0 8
4 6 7

1 2 3
0 5 8
4 6 7

1 2 3
4 5 8
0 6 7

1 2 3
4 5 8
6 0 7

1 2 3
4 5 8
6 7 0

1 2 3
4 5 0
6 7 8

1 2 3
4 0 5
6 7 8

States on optimal path: 14
Time taken (ms): 6.982
```

```
The total number of states explored: 156
For h3:
PASS
1 2 3
4 5 6
7 0 8

1 2 3
4 5 6
0 7 8

1 2 3
0 5 6
4 7 8

1 2 3
5 0 6
4 7 8

1 2 3
5 6 0
4 7 8

1 2 3
5 6 8
4 7 0

1 2 3
5 6 8
4 0 7

1 2 3
5 0 8
4 6 7

1 2 3
0 5 8
4 6 7

1 2 3
4 5 8
0 6 7

1 2 3
```

```
4 5 8
6 0 7

1 2 3
4 5 8
6 7 0

1 2 3
4 5 0
6 7 8

1 2 3
4 0 5
6 7 8

States on optimal path: 14
Time taken (ms): 2.991
The total number of states explored: 63
For h4:
PASS
1 2 3
4 5 6
7 0 8

1 2 3
4 5 6
0 7 8

1 2 3
0 5 6
4 7 8

1 2 3
5 0 6
4 7 8

1 2 3
5 6 0
4 7 8

1 2 3
5 6 8
4 7 0
```

```
1 2 3
5 6 8
4 0 7

1 2 3
5 0 8
4 6 7

1 2 3
0 5 8
4 6 7

1 2 3
4 5 8
0 6 7

1 2 3
4 5 8
6 0 7

1 2 3
4 5 8
6 7 0

1 2 3
4 5 0
6 7 8

1 2 3
4 0 5
6 7 8

States on optimal path: 14
Time taken (ms): 13779.5
The total number of states explored: 173262
```

Report

Better Heuristics Expanding Fewer States: In general, better heuristics should expand fewer states because they provide a more informed estimate of the actual cost to reach the goal. This means that A* will prioritize exploring paths that are more likely to lead to the goal. Among the heuristics mentioned, h3 (sum

of Manhattan distances) is generally considered better than h2 (number of displaced tiles) since it captures more information about the distance between a tile's current position and its goal position.
Expansion of States by Heuristics: Yes, this is generally true. A better heuristic should always underestimate the actual cost to reach the goal. This implies that if h(n) is a better heuristic than h'(n), then h(n) <= h'(n) for all states n. Consequently, if h(n) is admissible, h'(n) is also admissible. This ensures that all states expanded by a better heuristic will also be expanded by an inferior heuristic.
Monotone Restrictions on Heuristics: A heuristic h(n) is monotone (or consistent) if for every state n and every successor m of n generated by any action a, the estimated cost of reaching the goal from n is no greater than the cost of getting to m plus the estimated cost from m to the goal:

$h(n) \le cost(n, a, m) + h(m)$

In other words, the estimated cost to reach the goal from a state n should not increase when moving to a successor state m via a specific action a. This ensures that the heuristic remains consistent throughout the search process.

Un-reachability and Proof: If a state is unreachable, it means there is no sequence of valid actions that can lead to that state from the initial state. In the context of the 8-puzzle problem, this might happen if the puzzle is unsolvable (e.g., if there is an odd number of inversions in the initial configuration). In such cases, there's no way to reach the goal state, and thus, the unreachable state will not be expanded during the search. The proof relies on the properties of permutation parity and the mathematical properties of the 8-puzzle.
Monotone Restriction Verification: For a heuristic to follow the monotone restriction $h(n) \le cost(n, m) + h(m)$:

- a) For h2(n), number of displaced tiles, this condition holds. Moving a displaced tile closer to its goal position decreases h(n) while only incurring non-negative costs.
- b) For h3(n), sum of Manhattan distances, this condition also holds. Similar to h2, moving a tile towards its goal position reduces h(n) while incurring non-negative costs.

Violation of Monotonicity with Empty Tile Cost: If you add the cost of the empty tile (assuming it's treated as another tile) as part of the heuristic calculation, monotonicity will be violated. The reason is that moving the empty tile closer to

the goal might increase the heuristic value even if it lowers the actual cost, leading to inconsistent heuristic estimates.
Comparison of Heuristics:

- h1(n): This heuristic always returns 0, which is not informative at all. It’s similar to performing uniform cost search.
- h2(n): This heuristic considers only the number of misplaced tiles. It's admissible but not very informative, as it doesn't consider the relative positions of tiles.
- h3(n): This heuristic, the Manhattan distance sum, is admissible and more informative than h2. It considers the spatial relationship between tiles.
- h4(n): This heuristic is not admissible (per the problem statement) and could lead to suboptimal paths.

In terms of optimality, h3 provides the best estimate, followed by h2 and h1. h4 is not admissible and therefore cannot be relied upon for optimality.

Time complexity for A* depends on the quality of the heuristic. A more informative heuristic like h3 can potentially reduce the number of nodes expanded, improving the time complexity. h1 and h2 might expand more nodes, leading to a higher time complexity due to less informed decisions. h4's admissibility violation might lead to suboptimal solutions and hence unpredictable time complexity.

In conclusion, the quality of a heuristic significantly affects the performance of the A* algorithm in terms of optimality and time complexity. Admissible and informative heuristics tend to perform better by guiding the search towards the optimal solution more efficiently.