Jinwei Lin

Programming Assignment #1 Report

**Program Analysis**

**1. A\* search using the number of tiles out of place heuristic function**

|  |  |  |  |
| --- | --- | --- | --- |
| Input | # Of nodes expanded | Time required(s) | Sequence of moves |
| Easy | 6 | 0.000345 | Sequence of moves from start state to goal state:  1 3 4  8 6 2  7 0 5  1 3 4  8 0 2  7 6 5  1 3 4  8 2 0  7 6 5  1 3 0  8 2 4  7 6 5  1 0 3  8 2 4  7 6 5  1 2 3  8 0 4  7 6 5  Number of move is 5 | |
| Medium | 26 | 0.001161 | Sequence of moves from start state to goal state:  2 8 1  0 4 3  7 6 5  0 8 1  2 4 3  7 6 5  8 0 1  2 4 3  7 6 5  8 1 0  2 4 3  7 6 5  8 1 3  2 4 0  7 6 5  8 1 3  2 0 4  7 6 5  8 1 3  0 2 4  7 6 5  0 1 3  8 2 4  7 6 5  1 0 3  8 2 4  7 6 5  1 2 3  8 0 4  7 6 5  Number of move is 9 | |
| Hard | 71 | 0.004912 | Sequence of moves from start state to goal state:  2 8 1  4 6 3  0 7 5  2 8 1  4 6 3  7 0 5  2 8 1  4 0 3  7 6 5  2 8 1  0 4 3  7 6 5  0 8 1  2 4 3  7 6 5  8 0 1  2 4 3  7 6 5  8 1 0  2 4 3  7 6 5  8 1 3  2 4 0  7 6 5  8 1 3  2 0 4  7 6 5  8 1 3  0 2 4  7 6 5  0 1 3  8 2 4  7 6 5  1 0 3  8 2 4  7 6 5  1 2 3  8 0 4  7 6 5  Number of move is 12 | |
| Worst | N/A | N/A | N/A | |

**2. A\* search using the Manhattan heuristic function**

|  |  |  |  |
| --- | --- | --- | --- |
| Input | # of nodes expanded | Time required(s) | Sequence of moves |
| Easy | 5 | 0.000329 | Sequence of moves from start state to goal state:  1 3 4  8 6 2  7 0 5  1 3 4  8 0 2  7 6 5  1 3 4  8 2 0  7 6 5  1 3 0  8 2 4  7 6 5  1 0 3  8 2 4  7 6 5  1 2 3  8 0 4  7 6 5  Number of move is 5 | |
| Medium | 16 | 0.000752 | Sequence of moves from start state to goal state:  2 8 1  0 4 3  7 6 5  0 8 1  2 4 3  7 6 5  8 0 1  2 4 3  7 6 5  8 1 0  2 4 3  7 6 5  8 1 3  2 4 0  7 6 5  8 1 3  2 0 4  7 6 5  8 1 3  0 2 4  7 6 5  0 1 3  8 2 4  7 6 5  1 0 3  8 2 4  7 6 5  1 2 3  8 0 4  7 6 5  Number of move is 9 | |
| Hard | 25 | 0.001679 | Sequence of moves from start state to goal state:  2 8 1  4 6 3  0 7 5  2 8 1  4 6 3  7 0 5  2 8 1  4 0 3  7 6 5  2 8 1  0 4 3  7 6 5  0 8 1  2 4 3  7 6 5  8 0 1  2 4 3  7 6 5  8 1 0  2 4 3  7 6 5  8 1 3  2 4 0  7 6 5  8 1 3  2 0 4  7 6 5  8 1 3  0 2 4  7 6 5  0 1 3  8 2 4  7 6 5  1 0 3  8 2 4  7 6 5  1 2 3  8 0 4  7 6 5  Number of move is 12 | |
| Worst | 921 | 0.730357 | Sequence of moves from start state to goal state:  5 6 7  4 0 8  3 2 1  5 6 7  4 8 0  3 2 1  5 6 0  4 8 7  3 2 1  5 0 6  4 8 7  3 2 1  0 5 6  4 8 7  3 2 1  4 5 6  0 8 7  3 2 1  4 5 6  3 8 7  0 2 1  4 5 6  3 8 7  2 0 1  4 5 6  3 8 7  2 1 0  4 5 6  3 8 0  2 1 7  4 5 0  3 8 6  2 1 7  4 0 5  3 8 6  2 1 7  0 4 5  3 8 6  2 1 7  3 4 5  0 8 6  2 1 7  3 4 5  2 8 6  0 1 7  3 4 5  2 8 6  1 0 7  3 4 5  2 8 6  1 7 0  3 4 5  2 8 0  1 7 6  3 4 0  2 8 5  1 7 6  3 0 4  2 8 5  1 7 6  0 3 4  2 8 5  1 7 6  2 3 4  0 8 5  1 7 6  2 3 4  1 8 5  0 7 6  2 3 4  1 8 5  7 0 6  2 3 4  1 8 5  7 6 0  2 3 4  1 8 0  7 6 5  2 3 0  1 8 4  7 6 5  2 0 3  1 8 4  7 6 5  0 2 3  1 8 4  7 6 5  1 2 3  0 8 4  7 6 5  1 2 3  8 0 4  7 6 5  Number of move is 30 | |

**Questions:**

1. 9! = 362880 possible states

2. (2\*4 + 3\*4 + 4) / 9 = 2.67

3. Depends on the configuration of the worst-case problem.

4. b^d + b^(d-1) + b^(d-2) +…+ b^0

5. No. Tree depth is number of moves needed; in worst case it takes more than 50 moves. By using blind-search there are 2^50 + 2^49 +…+2^0 nodes to go over, so it would not terminate in one semester.

6. Because we tend to move the “1” into correct place, we can also observe whole board and avoid unnecessary moves. AI programs have specific algorithms to work out problems and stick to them, human performances are more flexible.

7. A\* is a best-first search which combines the features of uniform-cost search and heuristic search. It guarantees an optimal solution if the heuristic function is admissible. The drawback of A\* is the memory requirement, the entire open list must be saved.

Iterative deepening A\* combines iterative depth-first search and A\* search, it eliminates the memory constraints of A\*.

Depth-first branch-and-bound does not have the memory constraints like A\*, the search is efficient by keeping tack of the lowest-cost solution found so far.