CS 541 Programming Assignment 1 Write up

Korulkar Shweta

This programming assignment is the execution of 8 puzzle and 15 Puzzle(Bonus) using A* and Best-First Search algorithm with 3 different heuristics. The heuristics I used in the coding is 1) Misplaced tiles 2) Manhattan distance 3) Composite heuristic. The logic for misplaced tiles heuristic is, calculate the total number of tiles which are out of position. In Manhattan distance heuristic, calculate the sum of the distance of tiles from their goal position. In composite heuristic, I took the maximum of heuristic 1 i.e. Misplaced tile heuristic and heuristic 2 i.e. Manhattan distance heuristic. All these 3 heuristics are admissible heuristics so I choose these 3 heuristics. Below is the result of 8 puzzle using A* and Best-First Search algorithm.

- **8Puzzle using Best-First Search:**

*** Heuristic 1: Misplaced Tiles**

Path 1:

Initial State: (1 3 b 4 2 5 7 8 6)   Goal State: (1 2 3 4 5 6 7 8 b)

(1 3 b 4 2 5 7 8 6) —> (1 b 3 4 2 5 7 8 6) —> (1 2 3 4 b 5 7 8 6) —> (1 2 3 4 5 b 7 8 6) —> (1 2 3 4 5 6 7 8 b)

Number of steps: 4

Path 2:

Initial State: (1 3 4 8 6 2 7 b 5)   Goal State: (1 2 3 8 b 4 7 6 5)

(1 3 4 8 6 2 7 b 5) —> (1 3 4 8 b 2 7 6 5) —> (1 3 4 8 2 b 7 6 5) —> (1 3 b 8 2 4 7 6 5) —> (1 b 3 8 2 4 7 6 5) —> (1 2 3 8 b 4 7 6 5)

Number of steps: 5

Path 3:

Initial State: (1 2 3 b 4 6 7 5 8)   Goal State: (1 2 3 4 5 6 7 8 b)

(1 2 3 b 4 6 7 5 8) —> (1 2 3 4 b 6 7 5 8) —> (1 2 3 4 5 6 7 b 8) —> (1 2 3 4 5 6 7 8 b)—>

Number of steps: 3

Path 4:

Initial State: (b 3 4 1 8 2 7 6 5)   Goal State: (1 2 3 8 b 4 7 6 5)

(b 3 4 1 8 2 7 6 5) —> (1 3 4 b 8 2 7 6 5) —> (1 3 4 8 b 2 7 6 5) —> (1 3 4 8 2 b 7 6 5) —> (1 3 b 8 2 4 7 6 5) —> (1 b 3 8 2 4 7 6 5) —> (1 2 3 8 b 4 7 6 5)

Number of steps: 6

Path 5:

Initial State: (2 8 1 b 4 3 7 6 5)   Goal State: (4 2 8 3 b 1 7 6 5)

(2 8 1 b 4 3 7 6 5) —> (2 8 1 4 b 3 7 6 5) —> (2 8 1 4 3 b 7 6 5) —> (2 8 b 4 3 1 7 6 5) —> (2 b 8 4 3 1 7 6 5) —> (b 2 8 4 3 1 7 6 5) —> (4 2 8 b 3 1 7 6 5) —> (4 2 8 3 b 1 7 6 5)

Number of steps: 7

Average Number of steps: (4+5+3+6+7)/5 = 5

*** Heuristic 2: Manhattan distance**

Path 1:

Initial State: (1 3 b 4 2 5 7 8 6)   Goal State: (1 2 3 4 5 6 7 8 b)

(1 3 b 4 2 5 7 8 6) —> (1 b 3 4 2 5 7 8 6) —> (1 2 3 4 b 5 7 8 6) —> (1 2 3 4 5 b 7 8 6) —> (1 2 3 4 5 6 7 8 b)

Number of steps: 4

Path 2:

Initial State: (1 3 4 8 6 2 7 b 5)   Goal State: (1 2 3 8 b 4 7 6 5)

(1 3 4 8 6 2 7 b 5) —> (1 3 4 8 b 2 7 6 5) —> (1 3 4 8 2 b 7 6 5) —> (1 3 b 8 2 4 7 6 5) —> (1 b 3 8 2 4 7 6 5) —> (1 2 3 8 b 4 7 6 5)

Number of steps: 5

Path 3:

Initial State: (1 2 3 b 4 6 7 5 8)   Goal State: (1 2 3 4 5 6 7 8 b)

(1 2 3 b 4 6 7 5 8) —> (1 2 3 4 b 6 7 5 8) —> (1 2 3 4 5 6 7 b 8) —> (1 2 3 4 5 6 7 8 b)

Number of steps: 3

Path 4:

Initial State: (b 3 4 1 8 2 7 6 5)   Goal State: (1 2 3 8 b 4 7 6 5)

(b 3 4 1 8 2 7 6 5) —> (1 3 4 b 8 2 7 6 5) —> (1 3 4 8 b 2 7 6 5) —> (1 3 4 8 2 b 7 6 5) —> (1 3 b 8 2 4 7 6 5) —> (1 b 3 8 2 4 7 6 5) —> (1 2 3 8 b 4 7 6 5)

Number of steps: 6

Path 5:

Initial State: (2 8 1 b 4 3 7 6 5)   Goal State: (4 2 8 3 b 1 7 6 5)

(2 8 1 b 4 3 7 6 5) —> (2 8 1 4 b 3 7 6 5) —> (2 8 1 4 3 b 7 6 5) —> (2 8 b 4 3 1 7 6 5) —> (2 b 8 4 3 1 7 6 5) —> (b 2 8 4 3 1 7 6 5) —> (4 2 8 b 3 1 7 6 5) —> (4 2 8 3 b 1 7 6 5)

Number of steps: 7

Average Number of steps: (4+5+3+6+7)/5 = 5

*** Heuristic 3: Composite heuristic**

Path 1:

Initial State: (1 3 b 4 2 5 7 8 6)   Goal State: (1 2 3 4 5 6 7 8 b)

(1 3 b 4 2 5 7 8 6) —> (1 b 3 4 2 5 7 8 6) —> (1 2 3 4 b 5 7 8 6) —> (1 2 3 4 5 b 7 8 6) —> (1 2 3 4 5 6 7 8 b)

Number of steps: 4

Path 2:

Initial State: (1 3 4 8 6 2 7 b 5)   Goal State: (1 2 3 8 b 4 7 6 5)

(1 3 4 8 6 2 7 b 5) —> (1 3 4 8 b 2 7 6 5) —> (1 3 4 8 2 b 7 6 5) —> (1 3 b 8 2 4 7 6 5) —> (1 b 3 8 2 4 7 6 5) —> (1 2 3 8 b 4 7 6 5)

Number of steps: 5

Path 3:

Initial State: (1 2 3 b 4 6 7 5 8)   Goal State: (1 2 3 4 5 6 7 8 b)

(1 2 3 b 4 6 7 5 8) —> (1 2 3 4 b 6 7 5 8) —> (1 2 3 4 5 6 7 b 8) —> (1 2 3 4 5 6 7 8 b)

Number of steps: 3

Path 4:

Initial State: (b 3 4 1 8 2 7 6 5)   Goal State: (1 2 3 8 b 4 7 6 5)

(b 3 4 1 8 2 7 6 5) —> (1 3 4 b 8 2 7 6 5) —> (1 3 4 8 b 2 7 6 5) —> (1 3 4 8 2 b 7 6 5) —> (1 3 b 8 2 4 7 6 5) —> (1 b 3 8 2 4 7 6 5) —> (1 2 3 8 b 4 7 6 5)

Number of steps: 6

Path 5:

Initial State: (2 8 1 b 4 3 7 6 5)   Goal State: (4 2 8 3 b 1 7 6 5)

(2 8 1 b 4 3 7 6 5) —> (2 8 1 4 b 3 7 6 5) —> (2 8 1 4 3 b 7 6 5) —> (2 8 b 4 3 1 7 6 5) —> (2 b 8 4 3 1 7 6 5) —> (b 2 8 4 3 1 7 6 5) —> (4 2 8 b 3 1 7 6 5) —> (4 2 8 3 b 1 7 6 5)

Number of steps: 7

Average Number of steps: (4+5+3+6+7)/5 = 5

- **8Puzzle using A* search:**

*** Heuristic 1: Misplaced Tiles**

Path 1:

Initial State: (1 3 b 4 2 5 7 8 6)   Goal State: (1 2 3 4 5 6 7 8 b)

(1 3 b 4 2 5 7 8 6) —> (1 b 3 4 2 5 7 8 6) —> (1 2 3 4 b 5 7 8 6) —> (1 2 3 4 5 b 7 8 6) —> (1 2 3 4 5 6 7 8 b)

Number of steps: 4

Path 2:

Initial State: (1 3 4 8 6 2 7 b 5)   Goal State: (1 2 3 8 b 4 7 6 5)

(1 3 4 8 6 2 7 b 5) —> (1 3 4 8 b 2 7 6 5) —> (1 3 4 8 2 b 7 6 5) —> (1 b 4 8 3 2 7 6 5) —> (1 3 b 8 2 4 7 6 5) —> (1 b 3 8 2 4 7 6 5) —> (1 2 3 8 b 4 7 6 5)

Number of steps: 6

Path 3:

Initial State: (1 2 3 b 4 6 7 5 8)   Goal State: (1 2 3 4 5 6 7 8 b)

(1 2 3 b 4 6 7 5 8) —> (1 2 3 4 b 6 7 5 8) —> (1 2 3 4 5 6 7 b 8) —> (1 2 3 4 5 6 7 8 b)—>

Number of steps: 3

Path 4:

Initial State: (b 3 4 1 8 2 7 6 5)   Goal State: (1 2 3 8 b 4 7 6 5)

(b 3 4 1 8 2 7 6 5) —> (1 3 4 b 8 2 7 6 5) —> (1 3 4 8 b 2 7 6 5) —> (3 b 4 1 8 2 7 6 5) —> (1 3 4 8 2 b 7 6 5) —> (1 b 4 8 3 2 7 6 5) —> (1 3 b 8 2 4 7 6 5) —> (1 b 3 8 2 4 7 6 5) —> (1 2 3 8 b 4 7 6 5)

Number of steps: 8

Path 5:

Initial State: (2 8 1 b 4 3 7 6 5)   Goal State: (4 2 8 3 b 1 7 6 5)

(2 8 1 b 4 3 7 6 5) —> (2 8 1 4 b 3 7 6 5) —> (b 8 1 2 4 3 7 6 5) —> (2 8 1 7 4 3 b 6 5) —> (2 8 1 4 3 b 7 6 5) —> (2 b 1 4 8 3 7 6 5) —> (8 b 1 2 4 3 7 6 5) —> (2 8 b 4 3 1 7 6 5) —> (b 2 1 4 8 3 7 6 5) —> (2 b 8 4 3 1 7 6 5) —> (4 2 1 b 8 3 7 6 5) —> (b 2 8 4 3 1 7 6 5) —> (4 2 8 b 3 1 7 6 5) —> (4 2 8 3 b 1 7 6 5)

Number of steps: 13

Average number of steps: (4+6+3+8+13)/5 = 7

*** Heuristic 2: Manhattan distance**

Path 1:

Initial State: (1 3 b 4 2 5 7 8 6)   Goal State: (1 2 3 4 5 6 7 8 b)

(1 3 b 4 2 5 7 8 6) —> (1 b 3 4 2 5 7 8 6) —> (1 2 3 4 b 5 7 8 6) —> (1 2 3 4 5 b 7 8 6) —> (1 2 3 4 5 6 7 8 b)

Number of steps: 4

Path 2:

Initial State: (1 3 4 8 6 2 7 b 5)   Goal State: (1 2 3 8 b 4 7 6 5)

(1 3 4 8 6 2 7 b 5) —> (1 3 4 8 b 2 7 6 5) —> (1 3 4 8 2 b 7 6 5) —> (1 3 b 8 2 4 7 6 5) —> (1 b 3 8 2 4 7 6 5) —> (1 2 3 8 b 4 7 6 5)

Number of steps: 5

Path 3:

(1 2 3 b 4 6 7 5 8) —> (1 2 3 4 b 6 7 5 8) —> (1 2 3 4 5 6 7 b 8) —> (1 2 3 4 5 6 7 8 b)

Number of steps: 3

Path 4:

Initial State: (b 3 4 1 8 2 7 6 5)   Goal State: (1 2 3 8 b 4 7 6 5)

(b 3 4 1 8 2 7 6 5) —> (1 3 4 b 8 2 7 6 5) —> (1 3 4 8 b 2 7 6 5) —> (1 3 4 8 2 b 7 6 5) —> (1 3 b 8 2 4 7 6 5) —> (1 b 3 8 2 4 7 6 5) —> (1 2 3 8 b 4 7 6 5)

Number of steps: 6

Path 5:

Initial State: (2 8 1 b 4 3 7 6 5) Goal State: (4 2 8 3 b 1 7 6 5)

(2 8 1 b 4 3 7 6 5) —> (2 8 1 4 b 3 7 6 5) —> (2 8 1 4 3 b 7 6 5) —> (2 8 b 4 3 1 7 6 5) —> (2 b 8 4 3 1 7 6 5) —> (b 2 8 4 3 1 7 6 5) —> (4 2 8 b 3 1 7 6 5) —> (4 2 8 3 b 1 7 6 5)

Number of steps: 7

Average number of steps: (4+5+3+6+7)/5 = 5

*** Heuristic 3: Composite heuristic**

Path 1:

Initial State: (1 3 b 4 2 5 7 8 6) Goal State: (1 2 3 4 5 6 7 8 b)

(1 3 b 4 2 5 7 8 6) —> (1 b 3 4 2 5 7 8 6) —> (1 2 3 4 b 5 7 8 6) —> (1 2 3 4 5 b 7 8 6) —> (1 2 3 4 5 6 7 8 b)

Number of steps: 4

Path 2:

Initial State: (1 3 4 8 6 2 7 b 5) Goal State: (1 2 3 8 b 4 7 6 5)

(1 3 4 8 6 2 7 b 5) —> (1 3 4 8 b 2 7 6 5) —> (1 3 4 8 2 b 7 6 5) —> (1 3 b 8 2 4 7 6 5) —> (1 b 3 8 2 4 7 6 5) —> (1 2 3 8 b 4 7 6 5)

Number of steps: 5

Path 3:

(1 2 3 b 4 6 7 5 8) —> (1 2 3 4 b 6 7 5 8) —> (1 2 3 4 5 6 7 b 8) —> (1 2 3 4 5 6 7 8 b)

Number of steps: 3

Path 4:

Initial State: (b 3 4 1 8 2 7 6 5) Goal State: (1 2 3 8 b 4 7 6 5)

(b 3 4 1 8 2 7 6 5) —> (1 3 4 b 8 2 7 6 5) —> (1 3 4 8 b 2 7 6 5) —> (1 3 4 8 2 b 7 6 5) —> (1 3 b 8 2 4 7 6 5) —> (1 b 3 8 2 4 7 6 5) —> (1 2 3 8 b 4 7 6 5)

Number of steps: 6

Path 5:

Initial State: (2 8 1 b 4 3 7 6 5) Goal State: (4 2 8 3 b 1 7 6 5)

(2 8 1 b 4 3 7 6 5) —> (2 8 1 4 b 3 7 6 5) —> (2 8 1 4 3 b 7 6 5) —> (2 8 b 4 3 1 7 6 5) —> (2 b 8 4 3 1 7 6 5) —> (b 2 8 4 3 1 7 6 5) —> (4 2 8 b 3 1 7 6 5) —> (4 2 8 3 b 1 7 6 5)

Number of steps: 7

Average number of steps: (4+5+3+6+7) = 5

After looking at these results we can say that A* gives more optimal solution than Best-First Search. A* always explored the equal or more number of nodes than BFS. A* is optimally efficient for any given admissible heuristic. Here I used the tree search version of A* and Best-First Search algorithm. Using Best-First Search we can not guarantee to give solution in finite number of steps for some problems but A* gives the solution in finite steps. Best-First Search is not optimal and it's incomplete even in a finite number of steps. But A* is optimal, complete and optimally efficient. Best-First Search try to expand the node that is closest to the goal, on the ground that this is likely to lead to the solution quickly. For some problems Best-First Search will not give us solution but we will get solution using A*.

Now take a look at the result of 15Puzzle. The logic used for 15Puzzle is same as 8Puzzle. To solve 15Puzzle I used the same 3 heuristic mentioned above.

- **15Puzzle using Best-First Search:**

*** Heuristic 1: Misplaced Tiles**

1) Initial State: (1 2 3 4 5 6 7 8 9 b 10 12 13 14 11 15) Goal State: (1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 b)

Number of steps: 3

2) Initial State: (1 2 3 4 5 6 b 8 9 10 7 11 13 14 15 12) Goal State: (1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 b)

Number of steps: 3

3) Initial State: (1 b 3 4 5 2 6 8 9 10 7 12 13 14 11 15) Goal State: (1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 b)

Number of steps: 5

4) Initial State: (1 2 b 4 5 6 3 7 9 10 11 8 13 14 15 12) Goal State: (1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 b)

Number of steps: 4

5) Initial State: (1 2 3 4 5 6 7 8 9 10 b 11 12 13 14 15) Goal State: (1 2 3 4 5 6 7 8 9 10 14 11 12 b 13 15)

Number of steps: 2

6) Initial State: (1 6 2 4 9 5 3 7 13 10 11 8 b 14 15 12) Goal state: (1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 b)

Number of steps: 9

Average number of steps: (3+3+5+4+2+9)/6 = 4

*** Heuristic 2: Manhattan Distance**

1) Initial State: (1 2 3 4 5 6 7 8 9 b 10 12 13 14 11 15) Goal State: (1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 b)

Number of steps: 3

2) Initial State: (1 2 3 4 5 6 b 8 9 10 7 11 13 14 15 12) Goal State: (1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 b)

Number of steps: 3

3) Initial State: (1 b 3 4 5 2 6 8 9 10 7 12 13 14 11 15) Goal State: (1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 b)

Number of steps: 5

4) Initial State: (1 2 b 4 5 6 3 7 9 10 11 8 13 14 15 12) Goal State: (1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 b)

Number of steps: 4

5) Initial State: (1 2 3 4 5 6 7 8 9 10 b 11 12 13 14 15) Goal State: (1 2 3 4 5 6 7 8 9 10 14 11 12 b 13 15)

Number of steps: 2

6) Initial State: (1 6 2 4 9 5 3 7 13 10 11 8 b 14 15 12) Goal state: (1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 b)

Number of steps: No solution found

Average number of steps: (3+3+5+4+2)/5 = 3

*** Heuristic 3: Composite heuristic**

1) Initial State: (1 2 3 4 5 6 7 8 9 b 10 12 13 14 11 15) Goal State: (1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 b)

Number of steps: 3

2) Initial State: (1 2 3 4 5 6 b 8 9 10 7 11 13 14 15 12) Goal State: (1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 b)

Number of steps: 3

3) Initial State: (1 b 3 4 5 2 6 8 9 10 7 12 13 14 11 15) Goal State: (1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 b)

Number of steps: 5

4) Initial State: (1 2 b 4 5 6 3 7 9 10 11 8 13 14 15 12) Goal State: (1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 b)

Number of steps: 4

5) Initial State: (1 2 3 4 5 6 7 8 9 10 b 11 12 13 14 15) Goal State: (1 2 3 4 5 6 7 8 9 10 14 11 12 b 13 15)

Number of steps: 2

6) Initial State: (1 6 2 4 9 5 3 7 13 10 11 8 b 14 15 12) Goal state: (1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 b)

Number of steps: No solution found

Average number of steps: (3+3+5+4+2)/5 = 3

- **15Puzzle using A* search**

*** Heuristic 1: Misplaced tiles**

1) Initial State: (1 2 3 4 5 6 7 8 9 b 10 12 13 14 11 15) Goal State: (1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 b)

Number of steps: 3

2) Initial State: (1 2 3 4 5 6 b 8 9 10 7 11 13 14 15 12) Goal State: (1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 b)

Number of steps: 3

3) Initial State: (1 b 3 4 5 2 6 8 9 10 7 12 13 14 11 15) Goal State: (1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 b)

Number of steps: 5

4) Initial State: (1 2 b 4 5 6 3 7 9 10 11 8 13 14 15 12) Goal State: (1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 b)

Number of steps: 4

5) Initial State: (1 2 3 4 5 6 7 8 9 10 b 11 12 13 14 15) Goal State: (1 2 3 4 5 6 7 8 9 10 14 11 12 b 13 15)

Number of steps: 2

6) Initial State: (1 6 2 4 9 5 3 7 13 10 11 8 b 14 15 12) Goal state: (1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 b)

Number of steps: 9

Average number of steps: (3+3+5+4+2+9)/6 = 4

*** Heuristic 2: Manhattan Distance**

1) Initial State: (1 2 3 4 5 6 7 8 9 b 10 12 13 14 11 15) Goal State: (1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 b)

Number of steps: 3

2) Initial State: (1 2 3 4 5 6 b 8 9 10 7 11 13 14 15 12) Goal State: (1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 b)

Number of steps: 3

3) Initial State: (1 b 3 4 5 2 6 8 9 10 7 12 13 14 11 15) Goal State: (1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 b)

Number of steps: 5

4) Initial State: (1 2 b 4 5 6 3 7 9 10 11 8 13 14 15 12) Goal State: (1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 b)

Number of steps: 4

5) Initial State: (1 2 3 4 5 6 7 8 9 10 b 11 12 13 14 15) Goal State: (1 2 3 4 5 6 7 8 9 10 14 11 12 b 13 15)

Number of steps: 2

6) Initial State: (1 6 2 4 9 5 3 7 13 10 11 8 b 14 15 12) Goal state: (1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 b)

Number of steps: 16

Average number of steps: (3+3+5+4+2+16)/6 = 5

*** Heuristic 3: Composite heuristic**

1) Initial State: (1 2 3 4 5 6 7 8 9 b 10 12 13 14 11 15) Goal State: (1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 b)

Number of steps: 3

2) Initial State: (1 2 3 4 5 6 b 8 9 10 7 11 13 14 15 12)   Goal State: (1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 b)

Number of steps: 3

3) Initial State: (1 b 3 4 5 2 6 8 9 10 7 12 13 14 11 15)   Goal State: (1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 b)

Number of steps: 5

4) Initial State: (1 2 b 4 5 6 3 7 9 10 11 8 13 14 15 12)   Goal State: (1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 b)

Number of steps: 4

5) Initial State: (1 2 3 4 5 6 7 8 9 10 b 11 12 13 14 15)   Goal State: (1 2 3 4 5 6 7 8 9 10 14 11 12 b 13 15)

Number of steps: 2

6) Initial State: (1 6 2 4 9 5 3 7 13 10 11 8 b 14 15 12)   Goal state: (1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 b)

Number of steps: 16

Average number of steps: (3+3+5+4+2+16)/6 = 5

From above results we can say that, using A* we are getting optimal results. A* will give optimal and complete solution even for large problems. Best-First Search try to expand the node that is closest to the goal, on the ground that this is likely to lead to the solution quickly. But A* select the cheapest node from all expanded node so it will gives us a optimal and cheapest path from source to destination. For some initial states from where we reach to the goal state in more than 15 moves, we will get to see that A* will give more optimal solution by selecting minimum cost node as a next move than Best-First Search which use straight line distance heuristic to reach to goal state, which is not always optimal. Sometime we will not get solution in Best-First Search within finite steps.

*** Program (Python code) Execution Steps:**

Steps:

1) Execute the python script by typing " ./file_name.py " in terminal.

2) After executing the code, it will ask you to enter number 3 or 4. The code I did is for the N puzzle problem, so it will execute for any puzzle like 8puzzle and 15puzzle. To execute 8Puzzle we need to enter 3 as 8Puzzle takes initial and goal state as 3 * 3 matrix and enter 4 if you want want to execute 15Puzzle problem and enter 4 * 4 matrix.

3) Enter the initial and goal state numbers row wise separated by space. Enter each row in new line. Like for 8Puzzle, 1 3 b . Same pattern for 8puzzle goal state.

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

Similarly for 15Puzzle, enter 4*4 matrix as initial state and goal state. Enter each row in new line.