# Description of the heuristic and output generated while solving the 8-puzzle problem using A Star Search – Dhiraj Bag, 001911001033

Heuristic 1. Misplaced Tiles:

h1(n) = no of misplaced tiles, g(n) = no of level

so heuristic function is h1(n) + g(n)

Start state:

1 2 3

4 8 0

7 6 5

Goal state:

1 2 3

4 5 6

7 8 0

In level 0 we can move left, up or down.

1 2 3 1 2 0 1 2 3

4 0 8 4 8 3 4 8 5

7 6 5 7 6 5 7 6 0

For step 3 h1(n) = 3 and g(n) = 1. It has min heuristic cost than other moves so we have to choose this state.

So, we have to calculate h1(n) and g(n) for each level and need to pick the state which have least heuristic cost.

If h1(n) = 0 then we have reached the goal state and terminate.

For implementation we can take a priority queue with a comparator which will give us the state which have min heuristic cost when we call peek() method.

Heuristic 2. Manhattan Distance:

h1(n) = Manhattan distance, g(n) = no of level

so heuristic function is h1(n) + g(n)

Start state:

1 2 3

4 8 0

7 6 5

Goal state:

1 2 3

4 5 6

7 8 0

In level 0 we can move left, up or down.

1 2 3 1 2 0 1 2 3

4 0 8 4 8 3 4 8 5

7 6 5 7 6 5 7 6 0

To calculate Manhattan distance we have to calculate absolute difference of row and column for each

state and add it.

For level 0, step3 Manhattan distance is = 1+2+1 = 4

And g(n) = 1

It’s the minimum heuristic value level 1.So we have to choose step 3

1 2 3

4 8 5

7 6 0

This way we have to calculate heuristic value at each level and choose the state which have least cost.

If h1(n) = 0 then we have reached the goal state and terminate.