8-Puzzle problem:

Eight puzzle problem is also known as N puzzle problem or sliding

puzzle problem. The puzzle can be solved by moving the tiles one by

one in the single empty space and thus achieving the Goal state. Instead

of moving the tiles in empty space we can visualize moving the empty

space in place of tile. It can be movied in four directions(Movement of

emty space)

1. Up
2. Down
3. Right
4. Left

The empty space cannot be moved diagonally and can take **only one step at a time.**

**All possible move of a Empty tile**

**o- Position** total possible moves are **(2)**, **x - position** total possible moves are **(3)** and

#-position total possible moves are (4)

A\* Algorithm-

* A\* algorithm is one of the best and popular used for path finding and graph traversals
* A lot of games and web-based maps use this algorithm for path finding and graph traversals
* It is essentially a best first search algorithm

**How does it works?**

The algorithm works as follows:

* It maintains a tree of paths originating at the start node.
* It extends those paths one edge at a time.
* It continues until its termination criterion is satisfied.

A\* Algorithm extends the path that minimizes the following function-

where,

* n is the last node on the path
* g(n) is the cost of the path from start node to node n
* h(n) is a heuristic function that estimates cost of the cheapest path from node n to the goal node.

Algorithm-

* The implementation of A\* Algorithm invovles maintaing of two lists- OPEN and CLOSED.
* Open contains those nodes that have been evaluated by the heuristic function but have not been expanded into successors yet
* Closed contains those nodes that have already been visited.

The algorithm works as follows:

**Step 1:**

* Define a list OPEN.
* Initially, OPEN consists solely of a single node, the start node S.

**Step 2:**

If the list is empty, return failure and exit.

**Step 3:**

* Remove node n with the smallest value of f(n) from OPEN and move it to list CLOSED.
* If node n is a goal state, return success and exit.

**Step 4:**

Expand node n.

**Step 5:**

* If any successor to n is the goal node, return success and the solution by tracing the path from goal node to S.
* Otherwise, go to Step-06.

**Step 6:**

For each successor node,

* Apply the evaluation function f to the node.
* If the node has not been in either list, add it to OPEN.

**Step 7:**

Go back to step 2.