**Programming Project 1**

**Solving the 8-puzzle using A\* algorithm**

**Project Team**

* Anagha Sarmalkar (Student ID: 801077504)
* Madhuri Pawle (Student ID: )

**1. Introduction**

**1.1 8-puzzle problem**

In the 8-puzzle problem we have 9 tiles - one blank tile and 8 tiles numbered from 1 to 8. The goal of the puzzle is to reach a particular configuration of the 8 tiles (goal state) by swapping the blank tile with a numbered tile.

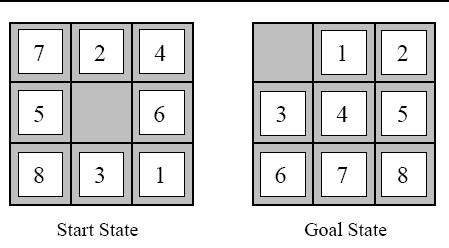


Fig 1.1 Example of 8-puzzle problem

**1.2 A\* Algorithm**

In this algorithm we decide to explore the node having the least f(n) value where f(n) is the sum of two parameters g(n) the step cost and h(n) the heuristic value and n being a particular node. We are using two heuristics here:

1. **Misplaced Tiles**

It gives the number of tiles that are not in the same position as in the goal state

1. **Manhattan Distance**

It gives the sum of number of steps needed for each tile to reach its position as in the goal state.

Min(f(n)), where f(n)= g(n)+h(n)

n= the node that is being explored

**2. Program Structure**

The project has been implemented using Python 3.7.0. The program consists of a class called A\_Star. The blank tile is represented using ‘0’.

**Class Variables**

1. i\_zero - Stores row value of zero in the state being explored
2. j\_zero - Stores column value of zero in the state being explored
3. step\_cost -Stores the g(n) value i.e. the step cost
4. nodes\_generated - Stores the count of total number of nodes generated
5. initial\_state - Stores the state being explored
6. initial\_state\_Copy - Stores a copy of initial state to use for different heuristic
7. goal\_state - Stores the goal state
8. unexplored\_states - Stores all the nodes that are yet to be explored
9. unexplored\_states\_node -Stores new nodes that have been generated in exploring\_nodes function
10. explored\_states - Stores list of nodes that has been explored
11. misplaced\_tiles\_list - Stores f(n) value corresponding to nodes in unexplored\_states list
12. manhattan\_distance\_list - Stores f(n) value corresponding to nodes in unexplored\_states list

**Functions**

1. **\_\_init\_\_():**

It is used to initialize the flag variable.

* **Local Variable**

**Flag** – It is used to decide which heuristic is being used misplaced tiles (flag =0) or manhattan distance (flag=1).

1. **finding\_zero():**

This function helps in finding the row and column of '0' in the state being explored and  
also checks if goal state is found. Also, few variables are initialized again so that can be used for a different heuristic.

* **Local Variable**

**a** – It is used to display the state as a 3X3 array.

1. **exploring\_nodes():**

This function finds all the possible nodes that can be generated by  
swapping the zero with numbers on top, right, bottom and left to create a new node  
and appends the new node to the unexplored\_states list

* **Local Variable**

**new\_state** – It stores the node generated by swapping the zero with any number on top, left, right and bottom.

1. **misplaced\_tiles():**

This function gives the number of tiles that are not in the same position in the state being explored as in goal state.

* **Local Variable**

**i,j,k –** They are used to traverse the unexplored\_ states\_node and goal\_state.

**count** – It is used to store the h(n) value.

**min\_misplaced** – It stores the minimum f(n) value from the misplaced\_tiles\_list.

1. **manhattan\_distance():**

This function gives the number of steps that are needed by each tile to reach the position as in goal state.

* **Local Variable**

**i, j, k –** They are used to access the numbered tile and traverse the unexplored\_ states\_node.

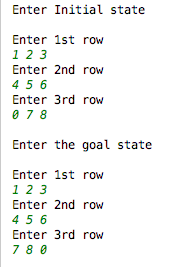
**gi, gj -** They are used to access the numbered tile in the goal state.

**distance** – It is used to store the h(n) value.

**min\_manhattan** – It stores the minimum f(n) value from the manhattan\_distance\_list.

**2.1 User Input**

The user has to give the initial state and the goal state as input. The input is being taken row-wise. It is taken as a group of three numbers, each number separated by a space.



**2.2 Functionality**

* We create a variable called A\_Star\_1 for the class A\_Star and call finding\_zero() function using A\_Star\_1 and passing flag as 0 which starts with misplaced tiles heuristic.
* The user is asked for input and the input is stored as a list of list.
* The finding\_zero() checks if the initial state is same as goal state if not finds the position of zero and calls the exploring node function
* The exploring\_node() function generates all possible nodes by swapping the ‘0’ with adjacent tiles.
* The generated nodes are added to unexplored\_states\_node which is appended to unexplored\_states list.
* The misplaced tile heuristic of every state in unexplored\_states is generated, added to step cost and appended to misplaced\_tiles\_list.
* The index of the minimum f(n) value is selected from the misplaced\_tiles\_list and the corresponding state from unexplored\_states is made as the initial state to repeat the above process.
* The flag is changed to 1 and all necessary variables are initialized in the finding\_zero() function.
* Once the goal state is found the control goes back to the beginning and A\_Star\_1 is used to call the finding\_zero() function again but this time with flag value as 1.
* The finding\_zero() checks if the initial state is same as goal state if not finds the position of zero and calls the exploring node function
* The exploring\_node() function generates all possible nodes by swapping the ‘0’ with adjacent tiles.
* The generated nodes are added to unexplored\_states\_node which is appended to unexplored\_states list.
* The manhattan distance heuristic of every state in unexplored\_states is generated, added to step cost and appended to manhattan\_distance\_list.
* The index of the minimum f(n) value is selected from the misplaced\_tiles\_list and the corresponding state from unexplored\_states is made as the initial state to repeat the above process.

**3. Sample Output**

**3.1 Sample 1**

****

**Misplaced Tiles**

Number of nodes generated 210 nodes

Number of nodes explored 119 nodes

**Manhattan Distance**

Number of nodes generated 18 nodes

Number of nodes explored 9 nodes

**3.2 Sample 2**



**Misplaced Tiles**

Number of nodes generated 22 nodes

Number of nodes explored 13 nodes

**Manhattan Distance**

Number of nodes generated 12 nodes

Number of nodes explored 6 nodes