

Programming assignment – 8 puzzle

This assignment consists in the resolution of an 8 puzzle (a 3x3 square with eight enumerated squares and one empty space). The objective of this task is to put all the squares sorted by number and the empty square on the leftmost upmost space (goal state) from any other initial state given in the code.

The algorithm used for the resolution of the problem is the A* algorithm, which means that when deciding which move to take we will select the state with the lower value of $f(x)$. This function is the sum of $g(x)$ (the cumulative value which represents how many moves has been done until that state) and $h(x)$ (the heuristic value in this problem given by the number of misplaced squares in the puzzle).

The algorithm has been implemented as a while loop and in each iteration it will first compare if it can do each move (up, down, left, right) and if it can it will save each new state on an array named as the direction. After that it will add all the elements that has not already been visited to the fringe. The fringe is represented as two arrays: one has all the values of $f(x)$ of each state and the other one contains each state as a tuple that contains the state and the sequence of actions the algorithm has taken until that same state. After that the algorithm moves to the next state choosing the minimum of the list of the fringe values and sets all the variables to the next state.

This algorithm is repeated afterwards but with heuristic value ($h(x)$) equal to 0, so it becomes an UCS algorithm. The result looks like this:

```
A*
The algorithm has found the solution in 4 iterations
The sequence of movements has been the following: ['L', 'U', 'L', 'U']
[[0, 1, 2], [3, 4, 5], [6, 7, 8]]
```

```
A* with  $h(x) = 0$  (UCS)
The algorithm has found the solution in 17 iterations
The sequence of movements has been the following: ['L', 'U', 'L', 'U']
[[0, 1, 2], [3, 4, 5], [6, 7, 8]]
```