

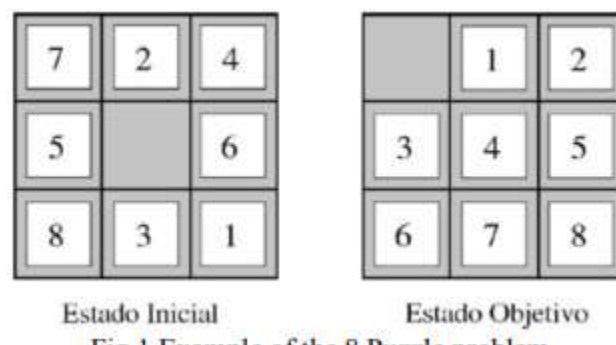
January 31 2019

A01630908 Adler Alonso Zamora Ruiz

INTELLIGENT SYSTEMS TC2011

UnInformed Search BFS

In this assignment we coded a BFS implementation where we need to solve a puzzle with a entry state and an expected state. The output of the program needs to have the path to the goal, the cost of the path, the number of visited nodes at the end (in Memory) and the running time in secs. Also de Memory used.



I have my program almost complete, I created a class with my array that represents the puzzle representation with an array being the entry state as [7, 2, 4, 5, 0, 6, 8, 3, 1] and the expected state as [0,1,2,3,4,5,6,7,8]. Also the class has the attribute parent to keep reference from which node each leaves comes from, and the index of the array where the zero is located (representing the empty space). I created the code of generating the next puzzle states according to the BFS algorithm and what it does, it created a queue first with the root puzzle class instance being enqueue and then dequeue to go according the logic. For each dequeue object we generate instances of the class puzzle with the movement (up, down, left, right) and look if we have it on our puzzleRegister, if not, we added it to it and queue it to our puzzle queue, we queue all of the valid new puzzle states, and Start again this process. For each movement applied to the current puzzle state we regenerate the puzzle zeroIndex, and when we find the expected state, we Will print the parent of each node interatively until we arrive to our root node.