

INSTRUCTIONS FOR RUNNING THE CODE

This project implements the generalized solver for an $n \times n$ bit puzzle using the Breadth First Search algorithm. It has a Node class having the attributes and methods useful for executing the BFS algorithm. Also, it keeps track of the parent node for backtracking after finding the goal state for the puzzle. It uses the deque class from collections module as the data structure to optimize the enqueue and dequeue operations. **NOTE: I have taken input test cases as row notation and solved the puzzle using that, but to display the output in the text files, I have used the column-wise notation.** Please follow the following steps to run the code:

1. Run the file `n*n_bit_solver.py` to execute the code for the 16 bit_puzzle. It has the list of all 5 test_cases and it prints the shortest path using backtracking for each test case.
2. Secondly, after running the python file in step 1, it will generate 5 output files (`nodePath_i.txt`) containing all the nodes from start to goal point for each test case in column-wise notation.
3. To run the code you will require the following modules:
 - a. time
 - b. copy
 - c. collections
 - d. numpy

4. Github link for reference:

https://github.com/savnani5/16_bit_puzzle/tree/BFS_with_backtrack

PS: Feel free to contact me if you have any difficulties in running the code. Thank you.