B551:Assignment_1

Part 1:

Search Algorithm #2 is employed to solve 16 puzzle.

- **State Space** : Every possible move given a state of the board.
- **Successor Function** : For every state there are 16 possible moves [L1 R1 L2 R2 L3 R3 L4 R4 U1 D1 U2 D2 U3 D3 U4 D4]. Thus the successor function generates 16 successors to any given state.
- The different heuristic functions used are:-
  1. **Circular Manhattan :** We calculate the Manhattan distance of each piece on the board with a slight modification. When we slide a row to left, the first piece in that row jumps 4 places to the right. We handle this case and count this jump as 1. To make this admissible we divide the entire sum by a factor 4 as in any move we end up moving 4 pieces of the puzzle. This is the main heuristic used in this code.
  2. **Manhattan Distance :** This is the normal Manhattan distance. We again divide the total sum of distances by 4 to make it admissible.
  3. **Reverse Permutation :** We count the number smaller numbers that appear ahead of given a number in the board.
  4. **No. of Rows/Columns in order :** Counted the number of rows or columns already in order. Although admissible, this heuristic didn't work well for the solver.

- Efforts to decrease the search space:-
  1. We find the rows and columns which are already in order , i.e: row1=[1,2,3,4], col1= [1,5,9,13]. For a given state we restrict the successor function to generate successors by moving these rows and columns.

- Even with the above tweeks, our program is not able to find solutions in reasonable amount of time to certain combinations of initial board. To tweek it further, we tried pre-calculating the states at depth 4 from the goal state itself. The number of states we had at depth 4, thus, were 16^4 . Our goal state changed to any one of the state in the above list. The time it takes to calculate states at depth 4 from the original goal state was around 15 sec. Apart from that, we were able to find the path from many states by adding the path we traversed to reach the intermediate goal state and reverse of the path to reach the intermediate goal state from the original goal state.