

## Read Me:

### To use the program for grading:

- Place the puzzle that must be solved into the 'puzzle.txt' file
- The output will list the set of moves required to solve the given puzzle
  - A move means that x switched position with the number in that direction
  - If it moves left, and the row looks like ['8', 'x', '7']
    - The result will be: ['x', '8', '7']
- The first output will be for Iterative Deepening Search
- The second output will be for Iterative Deepening A\* Search

### Heuristic Search:

The Heuristic for the Iterative Deepening A\* Search is based on the hamming distance. Not including x, I placed all values of the puzzle into a 1D array and compared it to the correct array. For each value in the wrong position, I increased the ham (representing the hamming distance) variable by 1.

### Comparison:

The iterative deepening A\* search performs better in terms of time and space compared to the iterative deepening search.

## Summary:

- Import copy is to create new instances of puzzle and not pointers
- Cutofflist is the list of f(n) in idas
- Puzzle class contains all functions needed for puzzle solving.
  - Makes it easier to pass to functions
- Goalstate returns what the problem should look like when solved
- AvailableMoves returns an array of the possible moves that can be played for an instance of a puzzle
- Left, right, up, and down shift the x value in that direction swapping what ever was in that place
- HammingDistance is the heuristic used for the iterative deepening A\* search
- ids is the iterative deepening search
- Idas is the iterative deepening A\* search
- Setup reads the file and creates required variables
- The rest executes the functions