Manan Agarwal
1BM 18CS05 2
AI Lab Test

Write up

goal state = [[1, 2, 3],
[4, 5, 6],
[7, 8, 0]]

start state = [[8, 2, 3],
[0, 4, 6],
[7, 5, 1]]

```
def distance(puzzle, item_to_calc, total_cal):
    t = 0
    for row in range(3):
        for col in range(3):
            val = puzzle.peek(row, col) - 1
            target_col = val % 3
            target_row = val / 3
            if target_row < 0:
                target_row = 2
            t += item_to_cal(row, target_row,
                             col, target_col)
    return total_cal(t)
```

```
def manhattan(puzzle):
    return distance(puzzle, lambda x,
        tx, ty, ty : abs(tx - x) + abs(ty - y),
        lambda t : t)

class SlidePuzzle():
    # class contains all functions for generating
    # and swapping
    def generate_sol_path(self, path):
        if self.parent is None:
            return path
        else:
            path.append(self)
            return self.parent.generate_
                sol_path(path)

    def generate_moves(self):
        free = self.get_legal_moves()
        zero = self.find(0)
```

```
def swap ( self , pos_first , pos_second ):
    temp = self . peek ( pos_first )
    self . poke ( pos_first [0] , pos_first [1] ,
                  self peek( pos_second ))
    self . poke ( pos_second [0] , pos_second [1] ,
                  temp )

def main ():
    p = SlidePuzzle ()
    p . startstate = [[8, 2, 3 ],[ 0, 4, 6 ],
                     [7, 5, 1 ]]

    print (p)
    path , count = p . solve ( manhattan )
    path . reverse ()
    for i in path :
        print (i)
```