8/10/24

Lab-02

①. Solving 8 puzzle problem using BFS algorithm

```
⇒ function BFS(start, goal)
        queue = empty dequeue()
        visited = empty set()
    parent-map = empty dictionary()

        queue.append(start)
        visited.add(start)
        parent-map[start] = None

    while queue is not empty:
        current = queue.popleft()
    if current is goal
        return current, parent-map

    for each neighbour in get-neighbour(current):
      if neighbour is not in visited:
            visited.add(neighbour)
            queue.append(neighbour)
            parent-map[neighbour] = current

    return none, parent-map

⇒ function get-neighbors(puzzle):
        neighbors = empty list
    (blank-x, blank-y) = find-blank(puzzle)

    moves = [(0,1), (0,-1), (1,0), (-1,0)]
```

```
for (dx, dy) in moves:
    nex_x = (blank_x + dx
    new_y = blank_y + dy

if (new_x, new_y) is within puzzle bounds:
    new_puzzle = copy(puzzle)
  swap new_puzzle [blank_x] [blank_y] with
        new_puzzle[new_x][new_y]
  neighbors.append (new_puzzle)

return neighbors

=> function find_blank (puzzle):
    for row in range (0,3)
        for col in range (0,3)
            if puzzle [row][col] == 0:
              return (row, col).

=> function get_solution_path(start, goal, parent_map
        path = empty_list
        current = goal

    while current is not None:
        path.append (current)
        current = parent_map[current]

    return reverse(path)
```

```
⇒ function solve_8_puzzle():
      print ("Rakhu input")

  start state = empty list
   for i in range(0,3):
      row = input
      start_state.append([int(x) for x in row.split()])

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

  print_puzzle(start_state)

  Solution, parent_map = BFS(start_state, goal_state)

   if solution is not None:
      print_puzzle(solution)

⇒ function print_puzzle(puzzle):
      for row in puzzle:
         print ()
```

Output:

```
Enter the starting state (use 0 for blank space)
Enter row1 : 1  2  3
Enter row2 :  5  4  6
Enter row3 :  0  8  7
Starting puzzle:              Solution found:
 1   2   3                     1   2   3
 5   4   6                     4   5   6
 —   8   7                     7   8   —
```