EXPERIMENT 6: Missionaries and Canabal problem

Aim: Implement an Algorithm in Python to solve the Missionaries and Canabal problem.

```
from collections import deque

def is_valid_state(state):
    missionaries_left, cannibals_left, boat_position = state
    missionaries_right = 3 - missionaries_left
    cannibals_right = 3 - cannibals_left

    if missionaries_left < cannibals_left and missionaries_left > 0:
        return False
    if missionaries_right < cannibals_right and missionaries_right > 0:
        return False
    return True
def generate_next_states(state):
    possible_moves = [(1, 0), (2, 0), (0, 1), (0, 2), (1, 1)]
    next_states = []
    for move in possible_moves:
        if state[2] == 1:  # boat on left bank
            new_state = (state[0] - move[0], state[1] - move[1], 0)
        else:  # boat on right bank
            new_state = (state[0] + move[0], state[1] + move[1], 1)
        if is_valid_state(new_state):
            next_states.append(new_state)
    return next_states

def bfs():
    initial_state = (3, 3, 1)
    goal_state = (0, 0, 0)
    visited = set()
    queue = deque([(initial_state, [])])

    while queue:
        current_state, path = queue.popleft()
        if current_state == goal_state:
            return path
        if current_state in visited:
            continue
        visited.add(current_state)
```

```
        for next_state in generate_next_states(current_state):
            queue.append((next_state, path + [next_state]))

    return None


def print_solution_path(path):
    if path:
        print("Solution Path:")
        for state in path:
            print(
                f"Missionaries: {state[0]}, Cannibals: {state[1]}, Boat position: {'Left' if state[2] == 1
else 'Right'}")
    else:
        print("No solution exists.")


# Solve the problem
solution_path = bfs()
print_solution_path(solution_path)
```

Output:

Missionaries: 3, Cannibals: 1, Boat position: Right

Missionaries: 4, Cannibals: 1, Boat position: Left

Missionaries: 3, Cannibals: 0, Boat position: Right

Missionaries: 3, Cannibals: 1, Boat position: Left

Missionaries: 1, Cannibals: 1, Boat position: Right

Missionaries: 2, Cannibals: 2, Boat position: Left

Missionaries: 0, Cannibals: 2, Boat position: Right

Missionaries: 0, Cannibals: 3, Boat position: Left

Missionaries: 0, Cannibals: 1, Boat position: Right

Missionaries: 1, Cannibals: 1, Boat position: Left

Missionaries: 0, Cannibals: 0, Boat position: Right

Result: Code has been Implemented successfully.