

**Code:**

```
import numpy as np
# Euclidean distance heuristic
def euclidean_distance(state, goal_state):
    total_distance = 0
    for i in range(3):
        for j in range(3):
            value = state[i, j]
            if value != 0:
                goal_pos = np.argwhere(goal_state == value)[0]
                distance = np.sqrt((i - goal_pos[0])**2 + (j - goal_pos[1])**2)
                total_distance += distance
    return total_distance

def generate_moves(state):
    moves = []
    empty_pos = np.argwhere(state == 0)[0]
    for move in [(0, 1), (0, -1), (1, 0), (-1, 0)]:
        new_pos = empty_pos + move
        if 0 <= new_pos[0] < 3 and 0 <= new_pos[1] < 3:
            new_state = state.copy()
            new_state[empty_pos[0], empty_pos[1]] = state[new_pos[0], new_pos[1]]
            new_state[new_pos[0], new_pos[1]] = 0
            moves.append(new_state)
    return moves

def steepest_ascent(initial_state, goal_state):
    current_state = initial_state.copy()
    while True:
        print("Current State:\n", current_state)
        current_distance = euclidean_distance(current_state, goal_state)
        best_next_state = None
        best_next_distance = float("inf")
        for move in generate_moves(current_state):
            move_distance = euclidean_distance(move, goal_state)
            if move_distance < best_next_distance:
                best_next_state = move
                best_next_distance = move_distance
        if best_next_distance >= current_distance:
            print("Reached local maximum. Final state:\n", current_state)
            break
        current_state = best_next_state
```

```
print("Enter the initial state:")
initial_state = np.array([list(map(int, input().split())) for _ in range(3)])

print("Enter the goal state:")
goal_state = np.array([list(map(int, input().split())) for _ in range(3)])
steepest_ascent(initial_state, goal_state)
```

## Output:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
● PS C:\Users\shrey\OneDrive\Desktop\programs> python -u "c:\Users\shrey\OneDrive\Desktop\programs\Python\AI3.py"
Enter the initial state:
2 8 3
1 6 4
7 0 5
Enter the goal state:
1 2 3
8 0 4
7 6 5
Current State:
[[2 8 3]
 [1 6 4]
 [7 0 5]]
Current State:
[[2 8 3]
 [1 0 4]
 [7 6 5]]
Current State:
[[2 0 3]
 [1 8 4]
 [7 6 5]]
Current State:
[[0 2 3]
 [1 8 4]
 [7 6 5]]
Current State:
[[1 2 3]
 [0 8 4]
 [7 6 5]]
Current State:
[[1 2 3]
 [8 0 4]
 [7 6 5]]
Reached local maximum. Final state:
[[1 2 3]
 [8 0 4]
 [7 6 5]]
○ PS C:\Users\shrey\OneDrive\Desktop\programs> █
```