Question:13

HILL CLIMBING ALGORITHM

AIM

To implement hill climbing algorithm using Python

ALGORITHM

1. Set the `current_state` to the `initial_state`.
2. Repeat until the `current_state` matches the `goal_state`:
   a. Generate neighboring states by adding or subtracting 1 from one of the coordinates of the `current_state`.
   b. Evaluate each neighboring state using the provided `evaluate` function.
   c. Select the neighboring state with the highest evaluation score as the `next_state`.
   d. If the evaluation score of the `next_state` is less than or equal to the evaluation score of the `current_state`, exit the loop.
   e. Update the `current_state` to the `next_state`.
3. Return the `current_state`.

CODE

```
import random
def hill_climbing(initial_state, goal_state, evaluate):
    current_state = initial_state
    while current_state != goal_state:
        neighbors = [(current_state[0] + random.choice([-1, 1]), current_state[1]), (current_state[0], current_state[1] + random.choice([-1, 1]))]
        next_state = max(neighbors, key=evaluate)
        if evaluate(next_state) <= evaluate(current_state):
            break
        current_state = next_state
    return current_state
initial_state = (0, 0)
goal_state = (5, 5)
evaluate = lambda state: -(abs(state[0] - goal_state[0]) + abs(state[1] - goal_state[1]))  # Example evaluation function
print(hill_climbing(initial_state, goal_state, evaluate))
```

OUTPUT

```
================= RESTART: C:\Users\Saaniya\Downloads\ai\13.py =================
(5, 5)
```