Assignment 3 (Hill Climbing)

Ayush Kumar Singh - 2211AI27

Gaurav Kumar - 2211AI28

Anurag Sharma - 2211AI29

A local search algorithm tries to find the optimal solution by exploring the states in the local region. Hill climbing is a local search technique that always looks for a better solution in its neighborhood.

a. Implement the **Hill Climbing Search Algorithm** for solving the 8-puzzle problem.
b. Check the algorithm for the following heuristics:
   i. h1(n) = number of tiles displaced from their destined position.
   ii. h2(n) = sum of the Manhattan distance of each tile from the goal position.

Instructions:

1. Input is given in a file in the following format. Read the input and store the information in a matrix. Configuration of the start state and the goal state can be anything. For example, given below, T1, T2, …, T8 are tile numbers, and B is blank space.

Start State

T3	T2	T1
T4	T5	T6
T8	T7	B

Goal State

T1	T2	T3
T4	T5	T6
T7	T8	B

Solution

Hill Climbing is heuristic search used for mathematical optimization problems in the field of Artificial Intelligence.

- Hill-climbing solves the problems where we need to maximize or minimize a given real function by choosing values from the given inputs. Example-8-puzzle problem.
- ‘Heuristic search’ means that this search algorithm may not find the optimal solution to the problem. However, it will give a good solution in a reasonable time.
- A heuristic function is a function that will rank all the possible alternatives at any branching step in the search algorithm based on the available information. It helps the algorithm to select the best route out of possible routes.

Algorithm for Simple Hill Climbing:

- **Step 1:** Evaluate the initial state, if it is goal state then return success and stop.
- **Step 2:** Loop Until a solution is found or there is no new operator left to apply.
- **Step 3:** Select and apply an operator to the current state.
- **Step 4:** Check new state:

    1. If it is goal state, then return success and quit.
    2. Else if it is better than the current state then assign new state as a current state.
    3. Else if not better than the current state, then return to step2.

- **Step 5:** Exit.

Start State

T3	T2	T1
T4	T5	T6
T8	T7	B

Goal State

T1	T2	T3
T4	T5	T6
T7	T8	B

Local Optimal State Reached

	Explored State	**Time Taken**
Tiles Displaced Heuristic	1	0.003028392791748047
Manhattan Heuristic	2	0.007494688034057617

Start State

B	T1	T3
T4	T2	T5
T7	T8	T6

Goal State

T1	T2	T3
T4	T5	T6
T7	T8	B

Global Optimal State Reached

	Explored State	**Time Taken**
Tiles Displaced Heuristic	5	0.012939929962158203
Manhattan Heuristic	5	0.011841773986816406