Step 1: Understand the Problem

The 8-Puzzle consists of a 3\times3 grid with 8 numbered tiles and one empty space (0).

Initial State: The starting arrangement of tiles.

Goal State: The desired final arrangement.

Moves: The empty space (0) can move Up, Down, Left, or Right by swapping with the adjacent tile.

Step 2: Understand Hill Climbing

Hill Climbing is a Local Search algorithm that moves continuously in the direction of increasing value (or decreasing cost) to find the solution.

Start with the Initial State.

Evaluate the current state using a Heuristic Function (h).

Generate all possible next moves of the empty tile (0).

Compare the neighbors:

If a neighbor has a better score than the current state, move to that state.

If no neighbor is better, the algorithm stops (this is a potential peak).

Step 3: The Heuristic Function (h)

To determine which move is "better," Hill Climbing uses a heuristic to estimate the distance to the goal:

Misplaced Tiles: The number of tiles currently in the wrong position.

Manhattan Distance: The sum of the vertical and horizontal distances each tile is from its goal position.

Step 4: Characteristics of Hill Climbing

Greedy Logic: It always chooses the immediate best neighbor without looking further ahead.

Memory Efficiency: It only stores the current state and its immediate neighbors; it does not keep a large queue of all discovered states.

Termination: The process stops when the Goal State is found or when no neighbor offers an improvement.

Not Complete: It can get stuck in "Local Maxima" where it thinks it has found the best state even if it is not the Goal State.

Step 5: Handling Obstacles

Unlike algorithms that explore every path, Hill Climbing can face specific issues:

Local Maxima: A state that is better than its neighbors but is not the goal.

Plateaus: A flat area of the search space where all neighbors have the same value.

Ridges: A slope that leads to a local peak rather than the highest goal point.

Step 6: Summary

Evaluate moves based on a heuristic value (h).

Move only to a state that improves the current score.

Requires very little memory compared to exhaustive search methods.

Stops when no better state is available.

Ideal for finding quick solutions in large search spaces, provided the heuristic is accurate.

Would you like me to demonstrate how to calculate the Manhattan Distance for the initial state provided in your document