1. Problem Description – 8-Puzzle

The 8-Puzzle problem is a classic artificial intelligence problem that consists of a 3×3 grid containing 8 numbered tiles (from 1 to 8) and one empty space (blank).

Objective:

The goal is to transform a given initial state into a goal state by sliding the tiles into the empty space.

Only one tile can be moved at a time, and moves are allowed up, down, left, or right.

Goal State (Typical):

1 2 3

4 5 6

7 8 _

Initial State Example:

1 2 3

4 _ 6

7 5 8

The problem is to find a sequence of valid moves that leads from the initial state to the goal state while respecting the movement constraints.\

2. Greedy Best-First Search Algorithm

Greedy Best-First Search (GBFS) is an informed search algorithm that selects which node to explore based only on a heuristic function.

Key Idea:

The algorithm always expands the node that appears to be closest to the goal, without considering the cost already taken to reach that node.

Evaluation Function:

f(n) = h(n)

Where:

h(n) is the heuristic value that estimates how close the current state is to the goal.

Heuristic Used:

In this project, we use the Manhattan Distance heuristic.

Manhattan Distance is calculated as the sum of the vertical and horizontal distances of each tile from its goal position (excluding the blank tile).

3.How Greedy Best-First Search Works (Step-by-Step)

1. Start from the initial state.
2. Calculate the heuristic value h(n) for the initial state.
3. Insert the state into a priority queue ordered by the heuristic value.
4. Remove the state with the lowest heuristic value from the queue.
5. If this state is the goal state, stop and return the solution.
6. Otherwise, generate all possible next states by moving the blank tile.
7. Calculate the heuristic for each new state and add them to the queue.
8. Repeat the process until the goal state is found or no states remain.

The algorithm is called “greedy” because it always chooses the state that looks best at the moment, without guaranteeing the optimal (shortest) solution.

Advantages and Disadvantages Advantages:

Simple to implement

Faster than uninformed search methods like BFS and DFS

Uses heuristic information to guide the search

Disadvantages:

Does not guarantee the shortest solution

Can get stuck exploring suboptimal paths

Performance depends heavily on the quality of the heuristic

4.Problem Solving of the 8-Puzzle Using Greedy Best-First Search

Step 1: Evaluate the Initial State

Tiles 5 and 8 are not in their correct positions.

The Manhattan Distance is calculated for each tile.

The initial heuristic value is greater than zero, meaning the state is not the goal.

Step 2: Generate Possible Moves

From the initial state, the empty space (_) can move:

Up

Down

Left

Right

Each move generates a new state.

Step 3: Heuristic Evaluation of Next States

For each generated state:

The Manhattan Distance is calculated.

The state with the lowest heuristic value is considered the best candidate.

Among all possible moves, the move that brings tile 5 closer to its goal position results in the smallest heuristic value.

Step 4: Select the Best State (Greedy Choice)

The algorithm chooses the state where tile 5 moves into the empty space:

1 2 3

4 5 6

7 _ 8

This state is closer to the goal because:

Tile 5 is now in its correct position.

The overall Manhattan Distance is reduced.

Step 5: Repeat the Process

From the new state:

Possible moves are evaluated again.

The heuristic values are recalculated.

The best move now is sliding tile 8 into the empty space.

Step 6: Reach the Goal State After

moving tile 8:

1 2 3

4 5 6

7 8 _

This matches the goal configuration.

Final Solution (Sequence of Moves)

1. Move tile 5 up into the empty space.
2. Move tile 8 left into the empty space.

The puzzle is solved in two moves.