CS571 AI LAB 03

11th September 2022

Ishita Singh 1901CS27
Kavya Goyal 1901CS30
Priyanka Sachan 1901CS43

Team Code: 1901cs30_1901cs27_1901cs43
https://colab.research.google.com/drive/1pKqmmp7ewR5kOeP1oV6o2gVh6A2NzQ-X?usp=sharing

OBJECTIVE

The objective of this assignment was to solve the 8-puzzle problem using Local Search Hill Climbing Algorithm

The following heuristic searches have been used this assignment

A. g(n) = least cost from source state to current state so far

B. Heuristics

1. h1(n) = number of tiles displaced from their destined position
2. h2(n) = sum of Manhattan distance of each tiles from the goal position

ALGORITHM

Classical hill climbing algorithm with 100 sideway moves is implemented.

- ➔ If current state is the target state, return true and print path
- ➔ If the heuristic function (cost) of all the valid neighbors is more than that of the current state, discard them and return current state as the local maxima
- ➔ If the heuristic cost of at least one neighbor is less than that of the current state, update this to be the next current state. If equal cost, check with respect to sideway move to check against flat or shoulder.

TEST CASES

```
Enter the start_state: 12345678B
Enter the target_state: 12345678B
Source State
[['1' '2' '3']
 ['4' '5' '6']
 ['7' '8' 'B']]
Target State
[['1' '2' '3']
 ['4' '5' '6']
 ['7' '8' 'B']]
+-------------+-------------+---------------+------------------+-------------+------------------+
| Algorithm   |   Path Cost |   Path States |   Execution Time | Reachable   | Path Traversed   |
+=============+=============+===============+==================+=============+==================+
| h1(n)       |           0 |             1 |      1.5974e-05  | True        | 12345678B        |
+-------------+-------------+---------------+------------------+-------------+------------------+
| h2(n)       |           0 |             1 |      1.88351e-05 | True        | 12345678B        |
+-------------+-------------+---------------+------------------+-------------+------------------+
```

```
Enter the start_state: B23145678
Enter the target_state: 2431B5678
Source State
[['B' '2' '3']
 ['1' '4' '5']
 ['6' '7' '8']]
Target State
[['2' '4' '3']
 ['1' 'B' '5']
 ['6' '7' '8']]
+-------------+-------------+---------------+------------------+-------------+---------------------------------+
| Algorithm   |   Path Cost |   Path States |   Execution Time | Reachable   | Path Traversed                  |
+=============+=============+===============+==================+=============+=================================+
| h1(n)       |           2 |             3 |      5.50747e-05 | True        | B23145678->2B3145678->2431B5678 |
+-------------+-------------+---------------+------------------+-------------+---------------------------------+
| h2(n)       |           2 |             3 |      8.22544e-05 | True        | B23145678->2B3145678->2431B5678 |
+-------------+-------------+---------------+------------------+-------------+---------------------------------+
```

```
Enter the start_state: 7B2453681
Enter the target_state: 12345678B
Source State
[['7' 'B' '2']
 ['4' '5' '3']
 ['6' '8' '1']]
Target State
[['1' '2' '3']
 ['4' '5' '6']
 ['7' '8' 'B']]
Stuck in local maxima for h(1)
Stuck in local maxima for h(2)
+-------------+-------------+---------------+------------------+-------------+-----------------------------------------+
| Algorithm   |   Path Cost |   Path States |   Execution Time | Reachable   | Path Traversed                          |
+=============+=============+===============+==================+=============+=========================================+
| h1(n)       |           0 |             4 |      0.000117064 | False       | 7B2453681->72B453681->72345B681->72345168B |
+-------------+-------------+---------------+------------------+-------------+-----------------------------------------+
| h2(n)       |           0 |             4 |      0.000944614 | False       | 7B2453681->72B453681->72345B681->72345168B |
+-------------+-------------+---------------+------------------+-------------+-----------------------------------------+
```

QUESTIONS

Make your observations for Hill climbing with respect to the time complexity (for near-optimal solution) and no of steps.

We iterate our algorithm to find the target state from 10^5 randomly generated grid states for h2 as the cost function over Hill Climbing Algorithm We find the average result

```
Success probability:  0.00145 %
Success Case:
Average states explored 11
Average time taken 0.0002438907753931333
Failure Case:
Average states explored 3
Average time taken 0.00045665136382205807
```

For near optimal solutions (or local maxima), we observe that the average number of steps taken to reach them is really less. The time complexity will directly be proportional to the number states explored.