

# **CS571 - ARTIFICIAL INTELLIGENCE LAB**

## **Lab - 3 Hill Climbing**

Jenish Monpara  
Tanishq Malu  
Tarusi Mittal

1901CS28  
1901CS63  
1901CS65

---

### **Instructions to run the code**

1. Launch terminal
2. Run "python3 main.py Start\_State.txt End\_State.txt"

**Sample Input-Output for each case for Success and failure cases:**

**Heuristic type 1: Number of tiles displaced from their destined position**

Solution exists case

**Start Case:**

B	T2	T3
T1	T4	T5
T6	T7	T8

**Goal Case:**

T2	T4	T3
T1	B	T5
T6	T7	T8

## Screenshot Of code:

```
(ML Course) jenish@Jenishs-MacBook-Pro lab3 % python3 main.py Start_State.txt End_State.txt
```

```
Enter the type of Hueristic function:
```

```
1. h1(n) = Number of tiles displaced from their destined position.
```

```
2. h2(n) = Sum of Manhattan distance of each tile from the goal
```

```
1
```

```
-----
```

```
-----
```

```
-----
```

```
Found a solution to the puzzle !
```

```
Start State of the Puzzle:
```

```
B T2 T3
```

```
T1 T4 T5
```

```
T6 T7 T8
```

```
Goal State of the Puzzle:
```

```
T2 T4 T3
```

```
T1 B T5
```

```
T6 T7 T8
```

```
Total number of states explored: 3
```

```
Total number of states to Optimal Path: 3
```

```
Optimal Path Cost: 2
```

```
Time taken: 0.0008342266082763672
```

```
B T2 T3
```

```
T1 T4 T5
```

```
T6 T7 T8
```

```
V
```

```
T2 B T3
```

```
T1 T4 T5
```

```
T6 T7 T8
```

```
V
```

```
T2 T4 T3
```

```
T1 B T5
```

```
T6 T7 T8
```

## Solution does not exists case:

### Start Case:

T2	T8	T3
T1	B	T5
T7	T6	T4

### Goal Case:

T2	T4	T3
T1	B	T5
T6	T7	T8

### Screenshot Of code:

```
(ML Course) jenish@Jenishs-MacBook-Pro lab3 % python3 main.py Start_State_2.txt End_State.txt
```

```
Enter the type of Hueristic function:
```

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

```
1
```

```
-----
```

```
00PS ! The program was unable to find a solution !
```

```
Start State of the Puzzle:
```

```
T2 T8 T3
```

```
T1 B T5
```

```
T7 T6 T4
```

```
Goal State of the Puzzle:
```

```
T2 T4 T3
```

```
T1 B T5
```

```
T6 T7 T8
```

```
Total number of states explored before ending the program: 1
```

**Heuristic type 2: Sum of Manhattan distance of each tiles from the goal position.**

Solution exists case

**Start Case:**

B	T2	T3
T1	T4	T5
T6	T7	T8

**Goal Case:**

T2	T4	T3
T1	B	T5
T6	T7	T8

**Screenshot Of code:**

(ML Course) jenish@Jenishs-MacBook-Pro lab3 % python3 main.py Start\_State.txt End\_State.txt

Enter the type of Hueristic function:

1.  $h1(n)$  = Number of tiles displaced from their destined position.
2.  $h2(n)$  = Sum of Manhattan distance of each tile from the goal

2

-----

-----

-----

Found a solution to the puzzle !

Start State of the Puzzle:

B T2 T3

T1 T4 T5

T6 T7 T8

Goal State of the Puzzle:

T2 T4 T3

T1 B T5

T6 T7 T8

Total number of states explored: 5

Total number of states to Optimal Path: 3

Optimal Path Cost: 2

Time taken: 0.0009918212890625

B T2 T3

T1 T4 T5

T6 T7 T8

V

T2 B T3

T1 T4 T5

T6 T7 T8

V

T2 T4 T3

T1 B T5

T6 T7 T8

Solution does not exists case

**Start Case:**

T2	T8	T3
T1	B	T5
T7	T6	T4

### Goal Case:

T2	T4	T3
T1	B	T5
T6	T7	T8

### Screenshot Of code:

```
(ML Course) jenish@Jenishs-MacBook-Pro lab3 % python3 main.py Start_State_2.txt End_State.txt
Enter the type of Hueristic function:
```

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

```
2
-----
-----
-----
-----
-----
```

```
00PS ! The program was unable to find a solution !
```

```
Start State of the Puzzle:
```

```
T2 T8 T3
T1 B T5
T7 T6 T4
```

```
Goal State of the Puzzle:
```

```
T2 T4 T3
T1 B T5
T6 T7 T8
```

```
Total number of states explored before ending the program: 5
```

-----END-----