**Assignment- cs561/571**

**Shashadhar Das**

**2111cs14**

**OUTPUTS :**

**(i)** The success or failure message :

**(ii)** Heuristics chosen, Start state and Goal state :

**(iii)** Total number of states explored :

**(iv)** Total number of states explored :

**(v)**  Total amount of time taken :

**Ans: The outputs are copied from the program after execution**

**Start Node:**

**1 2 3**

**4 5 6**

**7 0 8**

**------**

**Goal Node:**

**1 2 3**

**4 5 6**

**7 8 0**

**------**

**Considering blank tile**

**Heuristic choosen:Manhattan:**

**1 2 3**

**4 5 6**

**7 0 8**

**Node: 0**

**Depth: 0**

**Moves: []**

**------**

**[2]**

**1 2 3**

**4 0 6**

**7 5 8**

**Node: 1**

**Depth: 1**

**Moves: ['up']**

**------**

**1 2 3**

**4 5 6**

**0 7 8**

**Node: 2**

**Depth: 1**

**Moves: ['left']**

**------**

**1 2 3**

**4 5 6**

**7 8 0**

**Node: 3**

**Depth: 1**

**Moves: ['right']**

**------**

**Hill climbing is Success**

**Total no of state Explored: 3**

**Time: 0.0020689964294433594**

**------**

**Considering blank tile**

**Heuristic choosen:Misplaced tiles:**

**1 2 3**

**4 5 6**

**7 0 8**

**Node: 0**

**Depth: 0**

**Moves: []**

**------**

**[2]**

**1 2 3**

**4 0 6**

**7 5 8**

**Node: 1**

**Depth: 1**

**Moves: ['up']**

**------**

**1 2 3**

**4 5 6**

**0 7 8**

**Node: 2**

**Depth: 1**

**Moves: ['left']**

**------**

**1 2 3**

**4 5 6**

**7 8 0**

**Node: 3**

**Depth: 1**

**Moves: ['right']**

**------**

**Hill climbing is Success**

**Total no of state Explored: 3**

**Time: 0.0012249946594238281**

**------**

**------------------------------**

**Start Node:**

**1 2 3**

**4 5 6**

**7 0 8**

**------**

**Goal Node:**

**1 2 3**

**4 5 6**

**7 8 0**

**------**

**Not considering blank tile**

**Heuristic choosen:Manhattan:**

**1 2 3**

**4 5 6**

**7 0 8**

**Node: 0**

**Depth: 0**

**Moves: []**

**------**

**[1]**

**1 2 3**

**4 0 6**

**7 5 8**

**Node: 1**

**Depth: 1**

**Moves: ['up']**

**------**

**1 2 3**

**4 5 6**

**0 7 8**

**Node: 2**

**Depth: 1**

**Moves: ['left']**

**------**

**1 2 3**

**4 5 6**

**7 8 0**

**Node: 3**

**Depth: 1**

**Moves: ['right']**

**------**

**Hill climbing is Success**

**Total no of state Explored: 3**

**Time: 0.00146484375**

**------**

**Not considering blank tile**

**Heuristic choosen:Misplaced tiles:**

**1 2 3**

**4 5 6**

**7 0 8**

**Node: 0**

**Depth: 0**

**Moves: []**

**------**

**[1]**

**1 2 3**

**4 0 6**

**7 5 8**

**Node: 1**

**Depth: 1**

**Moves: ['up']**

**------**

**1 2 3**

**4 5 6**

**0 7 8**

**Node: 2**

**Depth: 1**

**Moves: ['left']**

**------**

**1 2 3**

**4 5 6**

**7 8 0**

**Node: 3**

**Depth: 1**

**Moves: ['right']**

**------**

**Hill climbing is Success**

**Total no of state Explored: 3**

**Time: 0.001459360122680664**

**------**

**------------------------------**

**Considering both Heuristic (h1 (n) + h2 (n))**

**Start Node:**

**1 2 3**

**4 5 6**

**7 0 8**

**------**

**Goal Node:**

**1 2 3**

**4 5 6**

**7 8 0**

**------**

**1 2 3**

**4 5 6**

**7 0 8**

**Node: 0**

**Depth: 0**

**Moves: []**

**------**

**Both: 4**

**[4]**

**1 2 3**

**4 0 6**

**7 5 8**

**Node: 1**

**Depth: 1**

**Moves: ['up']**

**------**

**1 2 3**

**4 5 6**

**0 7 8**

**Node: 2**

**Depth: 1**

**Moves: ['left']**

**------**

**1 2 3**

**4 5 6**

**7 8 0**

**Node: 3**

**Depth: 1**

**Moves: ['right']**

**------**

**Hill climbing is Success**

**Total no of state Explored: 3**

**Time: 0.0014798641204833984**

**------**

**1**

**(e). Constraints:**

**i)Check whether heuristics are admissible**

**Ans:** Yes, Heuristic are admissible.The Manhattan distance is an admissible heuristic in this

case because every tile will have to be moved at least the number of spots in between

itself and its correct position. The misplaced tiles also admissible heuristic.

**(ii). What happens if we make a new heuristics h3(n)= h1 (n) + h2 (n)?**

**Ans :**  h(n) = h1(n)+h2(n) is admissible, since given that h1(n) ≤ h ∗ (n) and h2(n)

≤ h ∗ (n) we deduce h1(n)+h2(n) 2 ≤ h ∗ (n) . In the output added above , we can see

there is no difference in the state explored.

**( iii). What happens if you consider the blank tile as another tile? (for each heuristic as**

**mentioned in d)**

**Ans.** The output is added above , it does not change anything if we consider the blank tile or

do not consider the black tile.

**(iv). What if the search algorithm got stuck into Local maximum? Is there any way to**

**get out of this?**

**Ans**. Backtracking technique can be a solution of the local maximum in state space

landscape. Create a list of the promising path so that the algorithm can

backtrack the search space and explore other paths as well.

**(v). What happens when all the neighbours of the current state have the same value?**

**How to get rid of this situation?**

**Ans :** The solution for the plateau is to take big steps or very little steps while

searching, to solve the problem. Randomly select a state which is far away

from the current state so it is possible that the algorithm could find

non-plateau region.