| **CS 571/561-ARTIFICIAL INTELLIGENCE LAB**  ENDSEM ASSIGNMENT: Hill Climbling |
| --- |

| Name | Sriram Pingali |
| --- | --- |
| Roll No | 1801CS37 |

### **Question - 1**

The heuristic *h1(n)* = **Admissible**.

* In the 8-puzzle problem, each displaced tile must be moved at least once to reach the goal state.
* So, the total number of moves to order the tiles correctly, or the cost to reach the goal state will be greater than or equal to the number of displaced tiles.
* Since, this heuristic is not overestimating the cost of reaching the goal state, it is admissible

The heuristic *h2(n)* = **Admissible**.

* Since we can only move one block at a time and in only one of the four directions. The optimal scenario for each block is that it has a clear, unobstructed path to its goal state. This is a Manhattan Distance.
* The rest of the states for a pair of blocks is sub-optimal, meaning it will take more moves than the Manhattan Distance to get the block in the right place.
* Thus, this heuristic does not overestimate the cost of reaching the goal state. Therefore it is admissible

### **Question - 2**

**Not Necessarily admissible.**

* A heuristic h is admissible if h(n) <= h(n)*.*

( *h*(n) = true cost to a nearest goal)

* We know that h1 and h2 are admissible. So, h1(n) <= h(n) and h2(n)<=h(n).

since,h3(n) = h1(n) + h2(n) does not guarantee that h3(n) <= h\*(n). Therefore, the admissibility of the heuristic h3(n) cannot be deduced.

### 

### **Question - 3**

For both the heuristics (h1 and h2) -> The Heuristic value would increase because, initially we do not consider the blank as a tile, but now the error associated with blank tile would also be considered. This might affect admissibility and the cost function value would increase.

### 

### **Question - 4**

Current states value is better than all neighboring states cost function values. As hill climbing method uses greedy approach it does not select any of the neighbouring states and terminate itself at that point.The process ends here even if an better solution exists.

**Ways to get out of the local optimum -**

* Backtracking- maintaining a list of visited states.If the current search state is an undesirable state then we can backtrack to one of the previous non-local optimum states and explore a new path.
* Random walk/random restart hill climbing

**Question - 5**

### If all neighbors have same value ,it is not possible to select the best direction.

### **To overcome :** Make a big jump. Randomly select a state far away from the current state. Chances are that we will land at a non-plateau region not same as current state.

### 