Question 1

part c

i) $f(n) = h(n) + g(n)$

$0 <= h(n) <= h^*(n)$

If the heuristic is $h(n) = g(n)$, this would be an ideal case as our algorithm would be moving on an optimal path.

If the heuristic is admissible $h(n) < g(n)$, A* provides the optimal path for the algorithm to take. The heuristic must not overestimate the effort required to achieve the goal in order to be considered admissible. (Underestimation)

If the heuristic function isn't admissible $h(n) > g(n)$, then it is possible to have an estimation that is larger than the actual path cost from some node to a goal node. (Overestimation).

Overall, the algorithm can select the optimal path thanks to the underestimate which happens when heuristic is admissible. However, overestimating the problem could lead the algorithm to choose certain undesirable paths.

ii) The monotonicity requirement of A* algorithm asks if an algorithm is “locally admissible.” That means for any two states, the heuristic function will be monotone if it underestimates the cost between these two states. In a search space, if a function is monotonic, it will always underestimate the cost of going between state i and state j. So, for state i and j, where j is the successor of state i;

$h(i) - h(j) <= cost(i, j)$

Here cost(i,j) is the actual cost it takes from i to j.

Along with this for a heuristic function to be monotone the heuristic value for the goal state should be equal to 0. For A* algorithm if the heuristic used is monotonic, the algorithm can be considered as optimal.