**Question 8a:** For a heuristic to be effective it must be both admissible, never overestimate, and consistent, the total cost only increases by the step cost of an action. I will also assume access to information regarding stations, station distances (cost) and location of the goal. With these stipulations the most admissible and consistent heuristic would be one that evaluates the start node and the goal state in order to find the minimum number of stops to that goal state. This heuristic is admissible because the total cost will always be the sum of step costs for each node to that goal, preventing overestimation. This heuristic is consistent because the path cost will only increment by the step cost at each node.

**Question 8b:** A heuristic needs to be both admissible and consistent. If we modify our A* algorithm heuristic to include a radius our heuristic is not admissible anymore. When using straight-line distance to our goal state, we are unable to overestimate the distance to our goal because our path will never be greater than the path it takes to get to the goal. This means we are unable to overestimate, ergo our heuristic is admissible. However, by expanding our goal to include any station within a certain radius of our goal, we can overestimate because the goal state our heuristic is based off of may be further away then a station that's closer to our initial state within the radius. This makes our heuristic inadmissible.

**Question 8c:** Simply put, if the distance cost between two linked stations is less than the calculated straight-line distance between the stations then we have overestimated in our heuristic making it inadmissible. We ensure admissibility and consistency in order to ensure that A* finds the most optimal path, however, when our heuristic is inadmissible it is unable to do this.

**Question 8d:** Modifying DFS to use iterative deepening would help the running time. This is ultimately dependent on the depth of the nodes, however, in the case of question 6 it would. This is because when using DFS the search algorithm will follow one branching path all the way to the furthest node which may be time and space complexity spent on a branching path that brings us nowhere close to our goal state. If we apply iterative deepening we can impose a depth limit which will allow us to use DFS to explore each node at each level. Question 6 adds the context of finding any station within a given distance from our goal station, meaning any station prior to our goal is also considered a target. This means we can find nodes within the radius of our goal state faster because they may be in a depth limit higher than the goal.