**CSE 4308-900 Artificial Intelligence**

**Task 2:**

**Breadth First search-** attached in the folder with the name Task\_2\_bfs.

**Depth First Search**- attached in the folder with the name Task\_2\_dfs.

**Uniform Cost search**- attached in the folder with the name Task\_2\_ucs.

**Iterative deepening search:**

Iteration 0:

Nodes Expanded- A

Nodes Added to the fringe- A

States added to the closed set- A

Iteration 1:

Nodes expanded- A

Nodes added to fringe- ADEB

Expand nodes D, E, B

States added to the closed set- ADEB

Iteration 2:

Nodes expanded- D

Nodes added to the fringe- G, E, E

Nodes expanded- G

Goal state found

States added to the closed set- ADEBG

**Task 3:**

1. Breadth first search, iterative deepening search, and uniform cost search.
2. Answer attached in the folder.
3. No, there is no one-to-one correspondence between nodes in the search tree and vertices in the search tree. The first level shows john having 3 adjacent nodes connected to it which shows that it is not one-to-one.
4. Answer attached in the folder.
5. Answer attached in the folder.
6. If every person has one node which takes up 1KB, 1 million people with 1 million nodes will take up 1 million KB. Since 1 million KB = 1 MB, the memory storage un worst case will not exceed 1 MB.

**Task 4:**

Heuristic 1: It is a non-admissible heuristic. To make it admissible, h(B) should be less than or equal to 45.

Heuristic 2: It is a non-admissible heuristic.

h(A) = less than or equal to 60

h(B) = less than or equal to 45

h(C) = less than or equal to 15

h(D) = 0

h(E) = less than or equal to 55

h(F) = less than or equal to 35

Heuristic 3: It is non-admissible.

h(C) = less than or equal to 15

h(D) = 0

Heuristic 4: It is admissible.

**Task 5:**

Image attached with the folder.

**Task 6:**

Figure 5:

For this figure, greedy search performs better than or the same as A\*.

In greedy search, f(n) = h(n), so the algorithm expands the nodes for the node with the smallest estimated distance to the goal. Since in this case, all adjacent nodes are connected to one another, there will not be an incomplete path.

A\* on the other hand has similar time complexity but its performance is degraded because it keeps all the nodes in memory. Overall, greedy will perform the same as or better than A\* search.

Figure 6:

Greedy search performs better at times, sometimes worse, and sometimes the same as A\* depending on the start and goal state.

For nodes that are not next to one another and reachable by a road, greedy search will perform better or same as A\* as stated for figure 5.

However, when two nodes are next to one another but don’t have a road connecting them, A\* will perform better. This is because A\* algorithm works by using a sorting criteria with f(n) = g(n) + h(n) which more accurately represents the actual road distance it would take to reach.