CSCI-404 Artificial Intelligence
Spring 2025, Homework Set 1

Question 1 Uninformed Search (20 points)

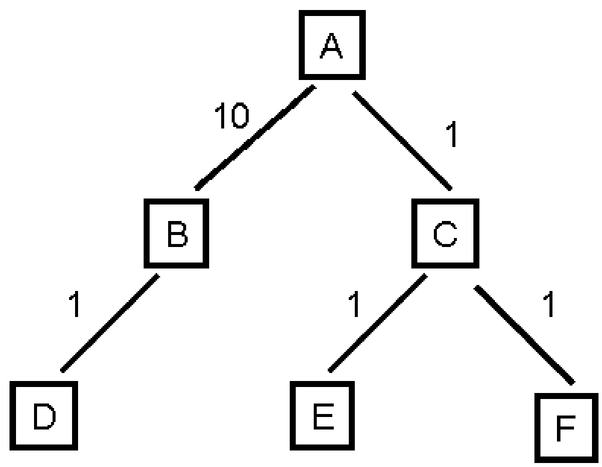

Fig. 1: An example search tree.

Consider the search tree shown in Figure 1. The number next to each edge is the cost to perform the action when the agent moves along that edge. Using the graph search methods listed below, list the order in which nodes will be visited (expanded, *i.e.*, removed from the fringe):

- breadth-first search (implemented by FIFO).
- depth-first search (implemented by LIFO).
- iterative deepening search.
- uniform cost search (implemented by a priority queue).

Note that revisiting the same state is ***not*** allowed in any iteration of the iterative deepening search and revisiting the same state is ***not*** allowed in the entire search process for the other search algorithms.

There is not a specific node corresponding to the goal state in the search tree. Your algorithm should conclude when all the nodes in the tree are visited.

There may exist more than one solution for each of the four search strategies due to randomness and uninformedness. Thus there could exist more than one correct answer for each question. You are not required to list all the correct answers. Instead, only one correct answer is required to get full credits.

Question 2 Search on Social Network Graphs (40 points)

A social network graph (SNG) is a graph where each vertex is a person and each edge represents an acquaintance. In other words, an SNG is a graph showing who knows who. For example, in the graph shown on Figure 2, George knows Mary and John, Mary knows Christine, Peter and George, John knows Christine, Helen and George, Christine knows Mary and John, Helen knows John, Peter knows Mary. The degrees of separation measures how closely connected two people are in the graph. For example, John has 0 degrees of separation from himself, 1 degree of separation from Christine, 2 degrees of separation from Mary, and 3 degrees of separation from Peter.

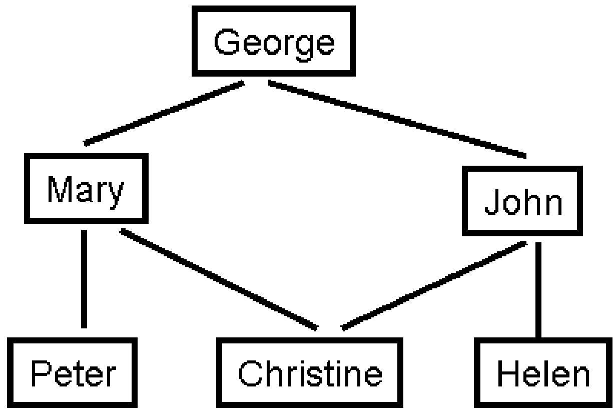

Fig. 2: A social network graph (SNG).

1. (10 points) From among breadth-first search, depth-first search, iterative deepening search, and uniform cost search, if tree search is being used, *i.e.*, there is no tracking on visited states, which one(s) guarantee finding the correct number of degrees of separation between any two people in the graph?
2. (10 points) For the SNG shown in Figure 2, draw the first three levels of the search tree, with John as the starting point (the first level of the tree is the root). Is there a one-to-one correspondence between nodes in the search tree and vertices in the SNG? Why, or why not? In your answer here, you should assume that the search algorithm does not try to avoid revisiting the same state.
3. (10 points) Draw an SNG containing exactly 5 people, where *exactly* two people (and no more) have 3 degrees of separation between them.
4. (10 points) Draw an SNG containing exactly 5 people, where all people have 1 degree of separation between them.

Question 3 Admissible Heuristics (10 points)

In an informed search algorithm, the heuristic function $h(n)$ is a (not necessarily accurate) heuristic estimate of the cost of the path from n to its nearest goal node, which have non-zero positive values for the states on a search graph. Suppose that the heuristic function $h(n)$ that we use for the A* search is $h(n) = 10$. In other words, according to this heuristic, the estimated cost of the path from EVERY node n to the goal is 10. We also know that every search tree in which we apply these implementations will have one and only one goal node.

1. (5 points) Is the heuristic function $h(n) = 10$ admissible? Please justify your answer.
2. (5 points) Will A*, using this heuristic function $h(n) = 10$, always find the smallest-cost solution? Please justify your answer.

Question 4 Informed Search (30 points)

Figures 3 and Figure 4 show maps where all the towns are on a grid. Each town T has coordinates (T_i, T_j), where T_i and T_j are non-negative integers. We use the term "*Euclidean distance*" for the straight-line distance between two towns. And we use the term "*driving distance*" for the length of the shortest driving route connecting two towns, also called as *"Manhattan distance"*. The only roads that exist connect towns that have Euclidean (straight-line) distance 1 from each other (however, there may be towns with Euclidean distance 1 from each other that are *NOT* directly connected by a road, for example in Figure 4).

Consider greedy search, where the node to be expanded is always the one with the shortest Euclidean distance to the destination. Also consider A* search, where $h(n)$ is the Euclidean distance from n to the destination. In A* search, the next node is picked not based on $h(n)$ but based on $f(n) = g(n) + h(n)$. **For each of the maps showing on Figure 3 and Figure 4, which of the following statements is true?**

- Greedy search always performs better than or the same as A*.
- Greedy search always performs worse than or the same as A*.
- Greedy search performs sometimes better, sometimes worse, and sometimes the same as A*, depending on the start and end states.

Here we assume that the searching agent can travel from any town on the map (say, starting at the one labeled as $(5, 4)$ or the one labeled as $(8, 8)$) to any other town (say, ending at the one labeled as $(2, 3)$ or the one labeled as $(0, 0)$) in the same map, as long as there exists a path to connect the two towns.

Please justify your answers. For the purposes of this question, the performance of a search algorithm is simply measured by the number of nodes visited (expanded, *i.e.*, removed from the fringe) by that algorithm. Please provide answers for Figure 3 and Figure 4 separately.

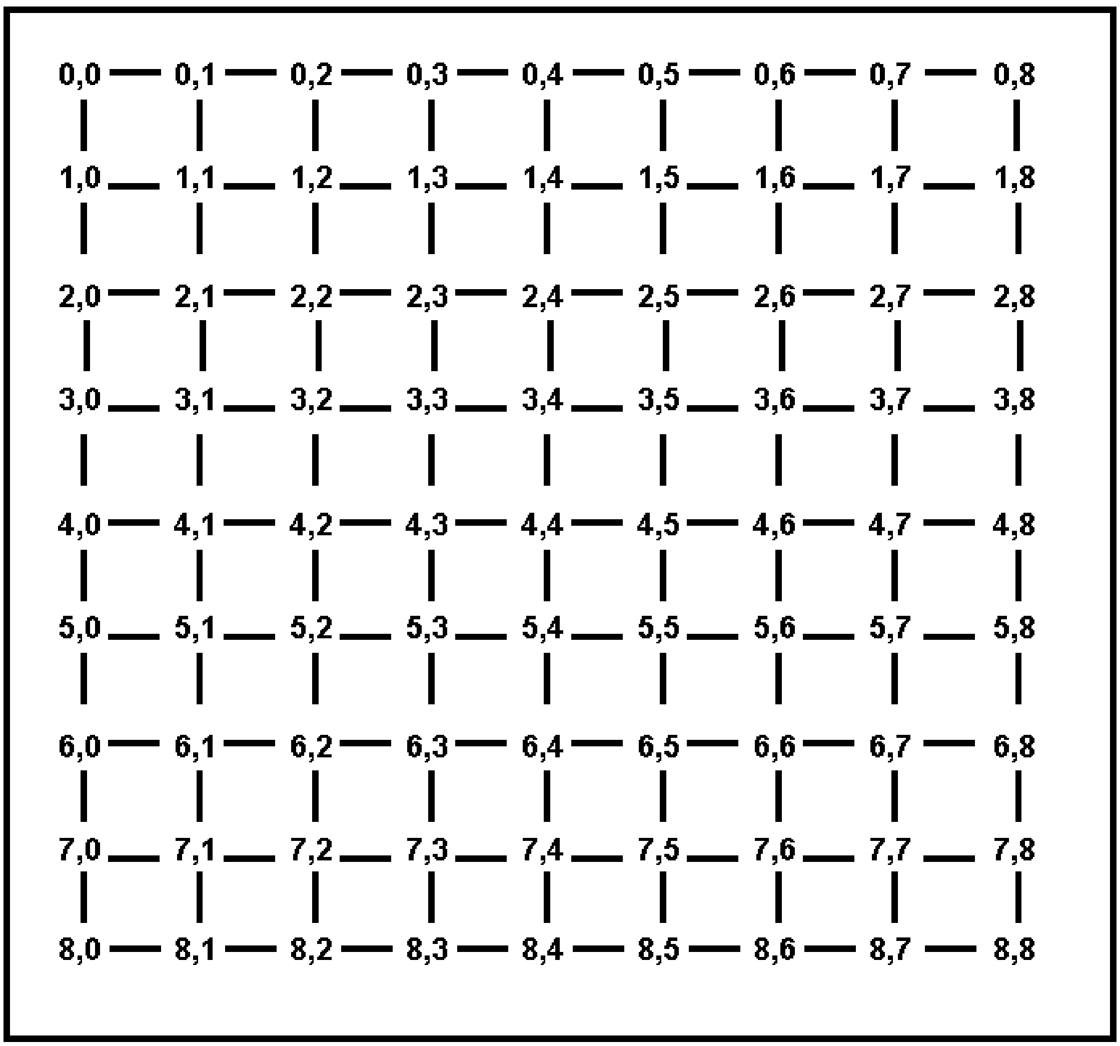

Fig. 3: A map of cities on a fully connected grid. Every city is simply named by its coordinates.

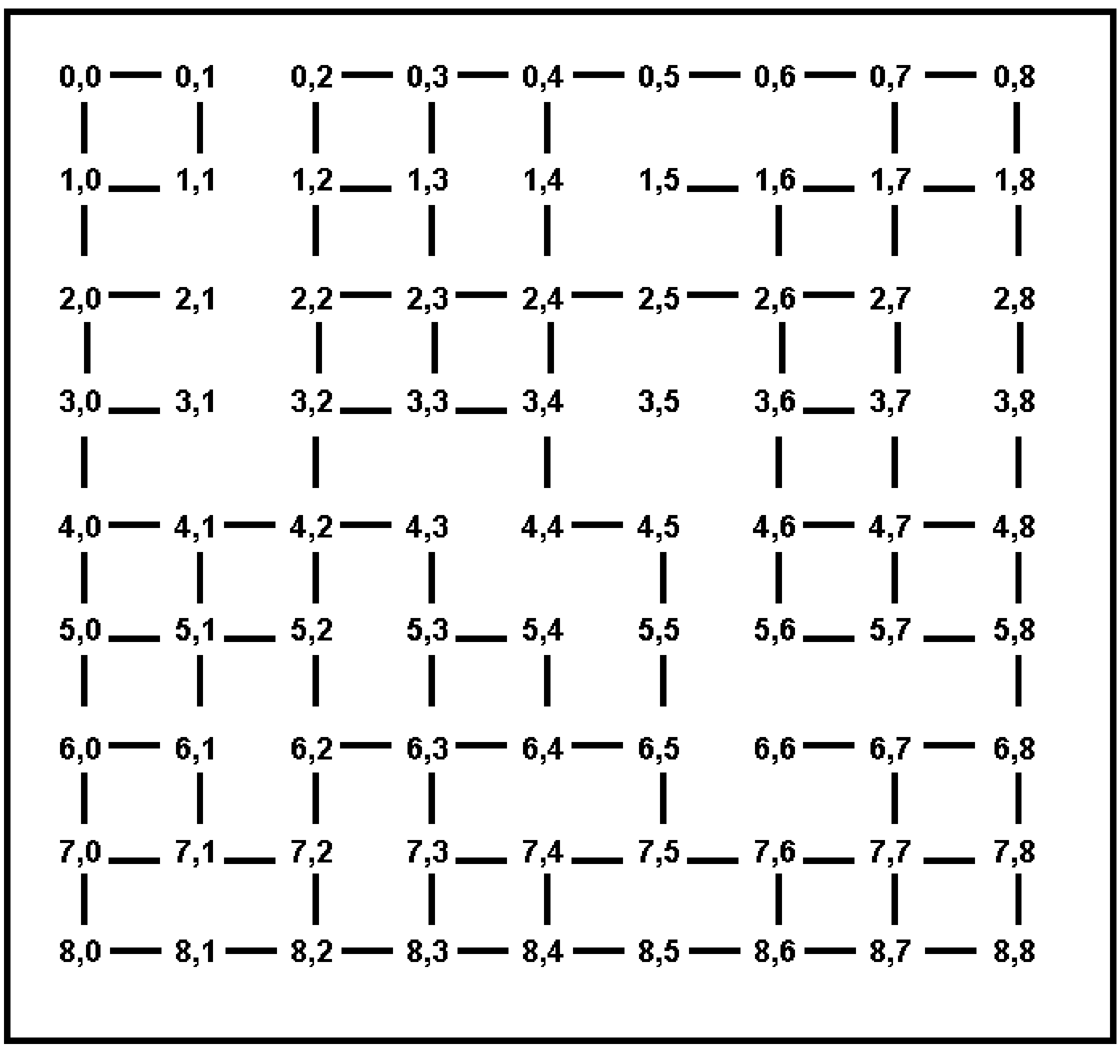

Fig. 4: A map of cities on a partially connected grid. Every city is simply named by its coordinates.