Question 1 (Agents) - 15 points

An agent lives in a grid world of size 10 x 10. The goal of the agent is to find a pot of gold. Only one pot of gold exists in the grid. At every step, the agent can move left, right, up, or down. The agent has two sensors:

- a sensor that detects the brightness level at the current square, and
- a sensor that detects if the current square is the square with the pot of gold or not.

The brightness level of a square is related to the location of the pot of gold as follows:

- Squares having a distance of 3 steps or less from the pot of gold have a brightness level of 10.
- All other squares have a brightness level of 0.

1a. (3 points) Is this environment deterministic or not? It is deterministic.

1b. (3 points) Is this environment fully observable or not? It is only partially observable.

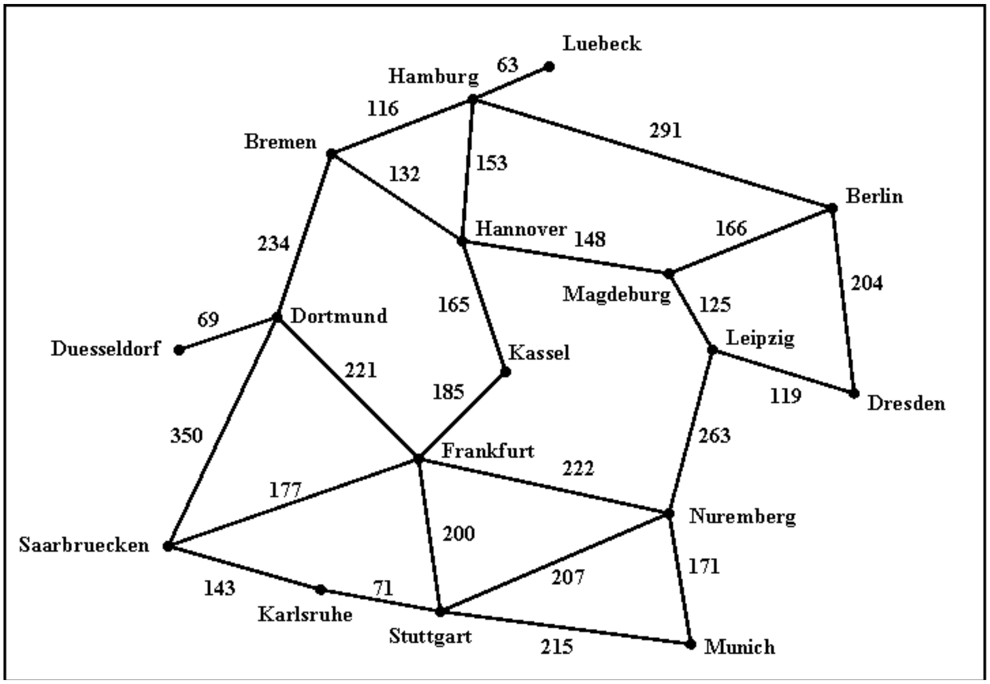
1c. (2 points) Is this environment static or not? It is static.

1d. (2 points) Is this environment discrete or not? It is discrete.

1e. (5 points) The agent determines the next move as follows: if there are adjacent positions that the agent has not visited yet, then it chooses randomly to move to one of those positions. Otherwise, it chooses randomly from all possible legal moves. Is this agent a reflex agent or not? Why?

No, it is not a reflex agent, because the next action is not determined by its sensory input, nor by a combination of sensory input plus memory of previous states.

Question 2 (Search) - 25 points



The above (hopefully familiar) map defines a search space, where each city is a state, and each road represents an action that can take the agent from one city to another. The cost of going through a road is simply the length of the road.

2a. (8 points) Given Hamburg as the start node and Munich as the goal node, draw the first three levels of the search tree (the root is the first level).

2b. (8 points) (NOTE: multiple answers may be correct here.) Assuming that the search does not keep track of cities already visited, list (in the order in which they are expanded) the first five nodes (including the start node corresponding to Hamburg) expanded by uniform cost search and iterative deepening search. For each of those first five nodes you just have to give the name of the corresponding city.

Uniform cost search:
Hamburg (cost 0), Luebeck (cost 63), Bremen (cost 116), Hamburg (cost 126), Hannover (cost 153)

Iterative deepening search:
Hamburg
Hamburg Luebeck Bremen Hannover Berlin

2c. (4 points, harder) For our route-finding search problem, precisely define an admissible heuristic function that makes A* behave exactly like uniform cost search (i.e., the sequence of nodes visited by A* is always the same as the sequence of nodes visited by uniform cost search, as long as there are no ties among nodes).

Hint 1: don't look for a smart heuristic here, just identify a function that is a legal admissible heuristic and that does what the question asks.

Hint 2: there is an answer that works for arbitrary search problems, and not just this specific search problem of finding routes in Germany.

Heuristic: h(n) = 0

2d. (5 points, harder) For our route-finding search problem, suppose that the search algorithm is given a pollution rating for each city on the map. Suppose that pollution ratings are related with distances to Munich as follows:

- any city within 300 km of driving distance from Munich has a pollution rating of 10.
- any city with driving distance from Munich between 301 km and 500 km has a pollution rating of 5.
- all other cities have a pollution rating of 0.

Using this information, **and assuming that Munich is always the goal node,** define a maximal admissible heuristic for A* (i.e., a heuristic that is not dominated by any other admissible heuristic that can be defined using this knowledge).

h(n) = 500 if polution rating of n is 0
h(n) = 300 if polution rating of n is 5
h(n) = 0 if polution rating of n is 10

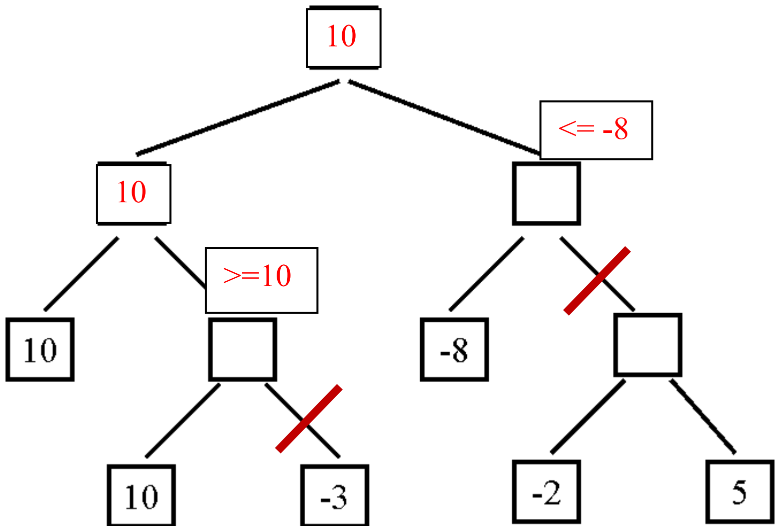
Question 3 (Games) - 30 points

3a. (10 points) Trace minimax (i.e., draw the search tree and show the utility value for each terminal and non-terminal node), starting at the following board state, and assuming that X makes the next move. Utility values are +1 if X wins, 0 for a tie, and -1 if O wins.

Starting board state:

X	X	O
X	O	O

3b. (10 points) In the search tree below, indicate what nodes will be pruned using alpha-beta search, and what the estimated utility values are for the rest of the nodes. Assume that, when given a choice, alpha-beta search expands nodes in a left-to-right order. Assume that the max player plays first.



3c. (10 points, harder) The search tree below is the same as the one shown for question 3b. For this question (question 3c only) we are given some additional knowledge about the game: the maximum utility value is 10, i.e., it is not mathematically possible for the MAX player to get an outcome greater than 10. How can this knowledge be used to further improve the efficiency of alpha-beta search? Indicate below the nodes that will be pruned using this improvement.