CS 4613 **Fall 2023**
HW 1 **E. K. Wong**

Total # of questions = 5. Total # of points = 85.

1. [10 points] Answer *True* or *False* to each of the following. No need to explain your answers.

- **(a)** One of the performance measures we use to evaluate a search algorithm is *completeness.* When we say an algorithm is *complete*, it means the algorithm is capable of generating a full search tree without missing any node.
- **(b)** The number of nodes generated by *breadth-first* search is the same as the number of nodes generated in the last iteration of the *iterative deepening* search. Assume that tree-like search (that allows repeated states) is used.
- **(c)** A *Utility-based agent* uses a function to evaluate how happy the agent will be in a certain state.
- **(d)** A *Model-based reflex agent* uses a world model to update its condition-action rules.

2. [30 points] In the vacuum cleaner problem as shown in Figure 1(a) below, there are four rooms and a vacuum cleaner. The floor of each room could be dirty or clean and the vacuum cleaner can be placed in any room initially. The vacuum cleaner can move to an adjacent room by performing a *left*, *right*, *up* or *down* action. It can also suck up dirt on the floor by performing a *suck* action. The symbols for the vacuum cleaner and dirt are as indicated in the figure. The goal is to clean up the dirt in all rooms.

a) What is the size of the state space in this problem?
b) How many states are reachable from the initial state shown in Figure 1(b) below? Include the initial state in the total number of reachable states. (There is no need to list all the reachable states or produce a search tree. Just determine the number of reachable states.)
c) Given the initial state as shown in Figure 1(b), use *breadth-first* search with *graph search* (does not allow repeated states) to find a solution to the problem. Draw the search tree produced and indicate the order of node expansion by putting a number in the upper-right corner of each node expanded. Highlight the solution path in the tree.
d) For the initial state as shown in Figure 1(b), can *depth-first* search with *graph search* be used to find a solution to the problem? Why or why not? Do not write more than six sentences. (No need to generate a search tree and find a solution. Just answer the question.)

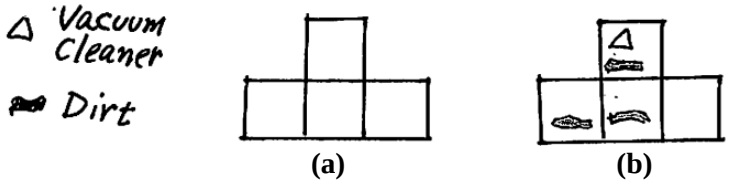


Figure 1. Vacuum cleaner problem.

3. [10 points] Consider a variant of the 8-puzzle problem. There are 14 tiles (numbered 1 to 14) and two blank positions on a 4×4 board. The goal is to slide the tiles from a given initial state to a desired goal state. A tile can slide into any of the two blank positions if it is horizontally or vertically adjacent to the blank position.

a) Define a set of actions for solving this problem.
b) What is the size of the state space for this problem? Assume that the two blank positions are undisguisable from each other in appearance.

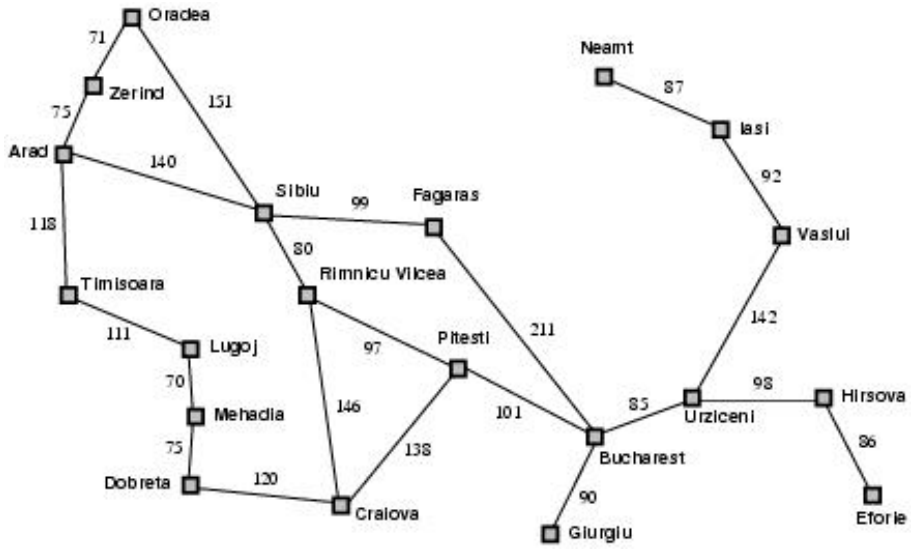
4. [15 points] An incremental formulation of the *4-queen* problem is as follows:

Incremental formulation

States: Any arrangement of j queens on a chessboard of size 4×4, for $j = 0$ to 4, and with one queen per **row** in the j **topmost rows**.
Initial state: No queen on the board.
Action: Add a queen to any square in the topmost empty row so that no two queens attack each other.
Goal: 4 queens are on the board, with no two queens attacking each other.

Find a solution to this problem by using the *depth-first* search algorithm with *tree-like* search (allows repeated states.) Draw the search tree produced and indicate the order nodes are **generated** by putting a number in the upper-right corner of the nodes. Note that with the formulation above, only nodes with legal states will be added to the search tree.

5. [20 points] Given the map of Romania in the figure below, the goal is to find the shortest path, in terms of distance travelled, between the start city of *Fagaras* and the goal city of *Oradea*. Use the *uniform-cost search* algorithm with *tree-like* search (allows repeated states) to find a solution. Draw the search tree produced and for each node *generated*, write the *path cost* to the left of the node. Also, indicate the order nodes are expanded by putting a number in the upper-right corner of each node expanded. Highlight the solution path in your tree. (When drawing your search tree, you may simply use the first letter to represent the name of the cities.)



Map of Romania. The numbers connecting cities represent distance in kilometers.