CS440/ECE448 Fall 2016 Midterm Review

You need to be able to define the following terms and answer basic questions about them:

- **Intro to AI, agents and environments**
    - Possible definitions of AI, pros and cons of each
    - Turing test: pros and cons, alternatives
    - Rationality
    - Utility, expected utility
    - PEAS
    - Environment characteristics: fully vs. partially observable, deterministic vs. stochastic, episodic vs. sequential, static vs. dynamic, discrete vs. continuous, single-agent vs. multi-agent, known vs. unknown

- **Search**
    - Search problem formulation: initial state, actions, transition model, goal state, path cost, state space
    - Tree search algorithm outline, frontier, search strategy, repeated state detection
    - Evaluation of search strategies: completeness, optimality, time complexity, space complexity
    - Uninformed search strategies: breadth-first search, uniform cost search, depth-first search, iterative deepening search
    - Informed search strategies: greedy best-first, A*, weighted A*
    - Heuristics: admissibility, consistency, dominance
    - Optimality of A*

- **Constraint satisfaction problems**
    - Backtracking search
    - Heuristics: most constrained/most constraining variable, least constraining value
    - Forward checking, constraint propagation, arc consistency
    - Tree-structured CSPs
    - Local search
    - SAT problem, NP-completeness

- **Games**
    - Zero-sum games
    - Game tree
    - Minimax strategy, minimax search
    - Alpha-beta pruning
    - Evaluation functions
    - Quiescence search, horizon effect
    - Monte Carlo tree search, AlphaGo
    - Stochastic games, expectiminimax
    - Partially observable games

- **Game theory**
  - Normal form representation
  - Dominant strategy
  - Nash equilibrium (pure and mixed strategy)
  - Pareto optimality
  - Examples of games: Prisoner's Dilemma, Stag Hunt, Game of Chicken
  - Mechanism design: auctions, regulation

- **Planning**
  - Situation space vs. plan space planners
  - Interleaved vs. non-interleaved planners
  - Partial order plan
  - Complexity of planning

Example test questions

1. Can an environment be both known and unobservable? Give an example.

2. What is the distinction between a world state and a search tree node?

3. In the tree search formulation, why do we restrict step costs to be non-negative?

4. Each of the following sub-problems describes a type of maze. For each type of maze, specify whether breadth-first-search (BFS) or depth-first-search (DFS) will more efficiently find a solution, and say why.

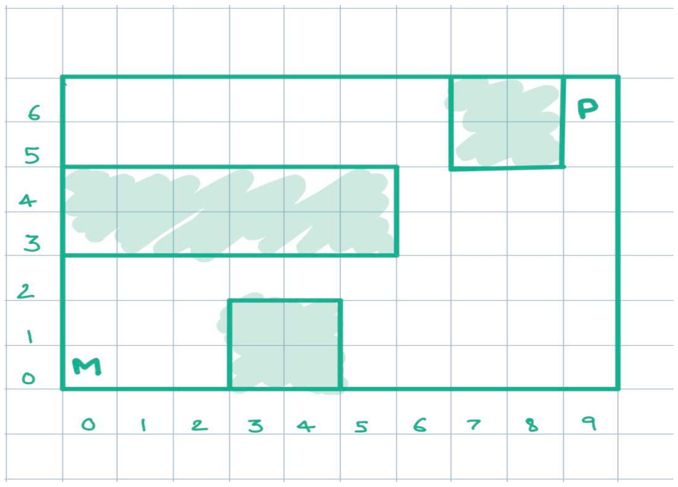
   a. The Albuquerque maze has 3 possible directions that you can take at each intersection. No path is longer than 25 steps. There is only one correct solution.
   b. The Belmont maze has 3 possible directions that you can take at each intersection. No path is longer than 25 steps. About half of all available paths are considered correct solutions to the maze.
   c. The Crazytown maze has 3 possible directions that you can take at each intersection. The maze is infinite in size, so some paths have infinite length. There is only one correct solution, which is known to require only 25 steps.

5. Suppose you are given a "perfect" heuristic function that gives the correct optimal distance from each node to the goal. Is greedy best-first search with this heuristic optimal? If not, give a counterexample.

6. Explain why it is a good heuristic to choose the variable that is *most* constrained but the value that is *least* constraining in a CSP search.

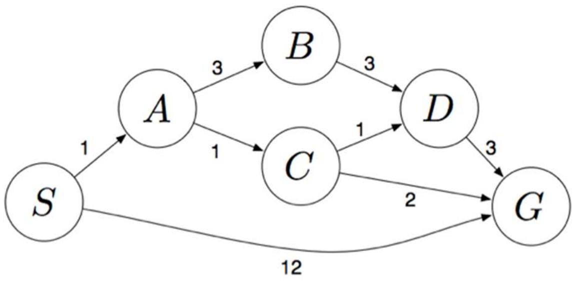
7. What is local search for CSPs? For which kinds of CSPs might local search be better than backtracking search? What about the other way around?

8. Refer to the maze shown below. Here, ‘M’ represents Mario, ‘P’ represents Peach, and the goal of the game is to get Mario and Peach to find each other. In each move, both Mario and Peach take turns. For example, *one* move would consist of Peach moving a block to the bottom from her current position, and Mario moving one block to the left from his current position. Standing still is also an option.



   a. Describe state and action representations for this problem.
   b. What is the branching factor of the search tree?
   c. What is the size of the state space?
   d. Describe an admissible heuristic for this problem.

9. Consider the search problem with the following state space:



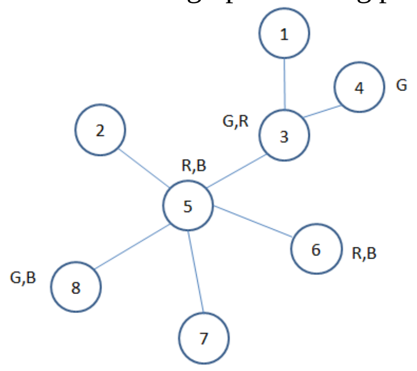
   S denotes the start state, G denotes the goal state, and step costs are written next to each arc. Assume that ties are broken alphabetically (i.e., if there are two states with equal priority on the frontier, the state that comes first alphabetically should be visited first).

   a. What path would BFS return for this problem?
   b. What path would DFS return for this problem?
   c. What path would UCS return for this problem?
   d. Consider the heuristics for this problem shown in the table below.

State	h_1	h_2
S	5	4
A	3	2
B	6	6
C	2	1
D	3	3
G	0	0

Is h_1 admissible? Is it consistent?
Is h_2 admissible? Is it consistent?

10. Consider the graph-coloring problem on the following tree-structured CSP:



We assume there are three available colors (R,G,B) and some nodes are constrained to use only a subset of these colors, as indicated above. Show all the steps for applying the tree-structured CSP algorithm for finding a solution to this problem.

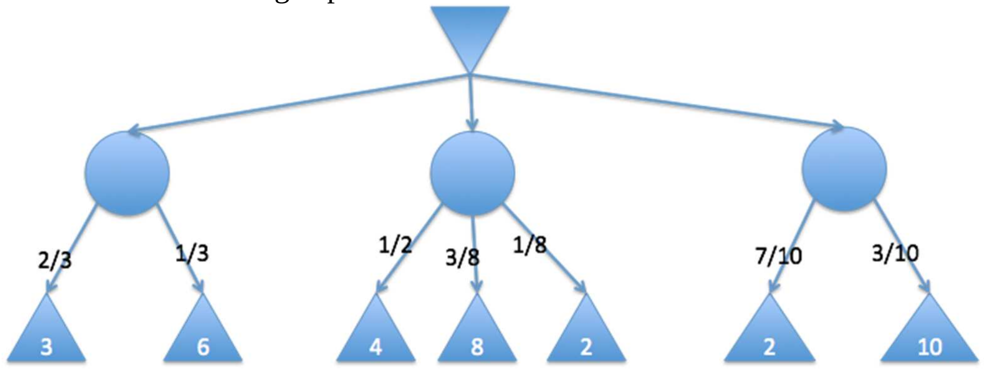
11. For each of the following problems, determine whether an algorithm to optimally solve the problem requires computation time that is polynomial or exponential in the parameters d and m.

    a. A map has d regions. Colors have been applied to all d regions, drawing from a set of m possible colors. Your algorithm needs to decide whether or not any two adjacent regions have the same color.
    b. A map has d regions. Your algorithm needs to assign colors to all d regions, drawing colors from a set of m possible colors, in order to guarantee that no two adjacent regions have the same color.
    c. Your algorithm needs to find its way out of a maze drawn on a d-by-d grid.
    d. Your algorithm needs to find the shortest path in a d-by-d maze while hitting m waypoints (equivalent to dots in MP1 part 1.2).
    e. Your algorithm needs to find the best strategy for a zero-sum game. There are two players. At each turn, each of the players chooses from among m possible moves. After d rounds of game play, the game ends.

12. How can randomness be incorporated into a game tree? How about partial observability (imperfect information)?

13. In the lectures, we covered Nash equilibrium strategies for simultaneous move games. We can also consider minimax strategies for such games, defined in the same way as for alternating games. What would be the minimax strategies in the Prisoner's Dilemma, Stag Hunt, and Game of Chicken? Do they differ from Nash equilibrium strategies? When/why would one prefer to choose a minimax strategy rather than a Nash equilibrium strategy?

14. Consider the following expectiminimax tree:



Circle nodes are chance nodes, the top node is a min node, and the bottom nodes are max nodes.

a. For each circle, calculate the node values, as per expectiminimax definition.
b. Which action should the min player take?

15. Suppose that both Alice and Bob want to go from one place to another. There are two routes R1 and R2. The utility of a route is inversely proportional to the number of cars on the road. For instance, if both Alice and Bob choose route R1, the utility of R1 for each of them is 1/2.

a. Write out the payoff matrix.
b. Is this a zero-sum game?
c. Find dominant strategies (if any).
d. Find pure strategy equilibria (if any).
e. Find the mixed strategy equilibrium.