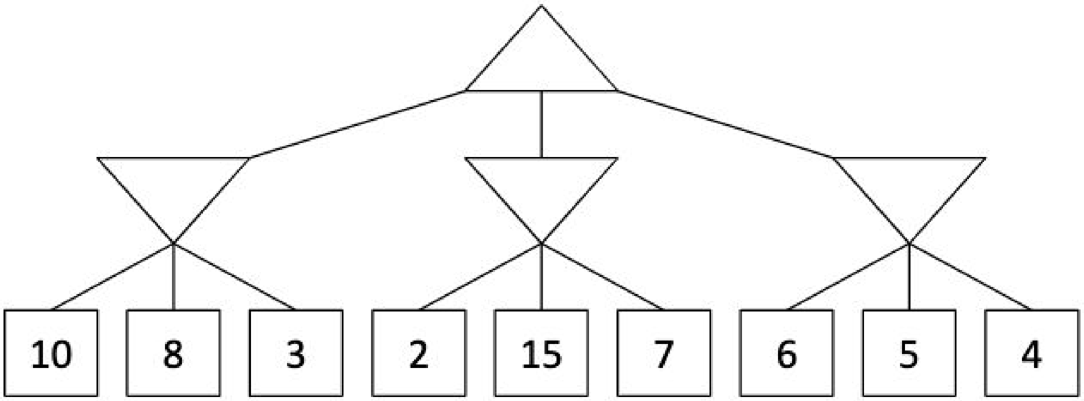
CS 3346A / CS 3121A Assignment 2

Due: Saturday, Nov. 16th, 2019 (midnight)
Total Weight: 15%
Silent Policy: A silent policy will take effect 24 hours before this assignment is due, i.e. no question about this assignment will be answered, whether it is asked on the discussion board, via email or in person.

Part 1*: 6%*

1 Games:



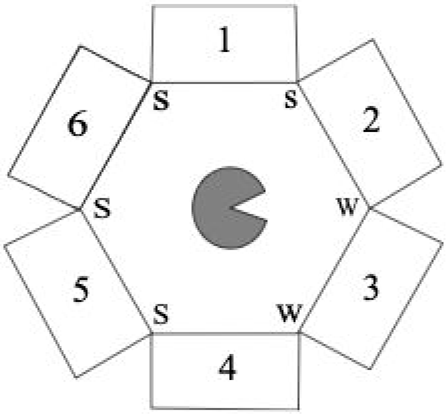
1. Consider the zero-sum game tree shown below. Triangles that point up, such as at the top node (root), represent choices for the maximizing player; triangles that point down represent choices for the minimizing player. Assuming both players act optimally, fill in the minimax value of each node.

2. Which nodes can be pruned from the game tree above through alpha-beta pruning? If no nodes can be pruned, explain why not. Assume the search goes from left to right; when choosing which child to visit
first, choose the left-most unvisited child.

3. Again, consider the same zero-sum game tree, except that now, instead of a minimizing player, we have a chance node that will select one of the three values uniformly at random. Fill in the expectimax value of each node. Redraw the game tree below to show your answer.

4. Which nodes can be pruned from the game tree above through alpha-beta pruning? If no nodes can be pruned, explain why not.

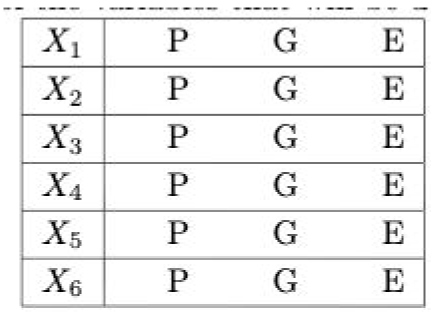
2 CSPs: Trapped Pacman



Pacman is trapped! He is surrounded by mysterious corridors, each of which leads to either a pit (P), a ghost (G), or an exit (E). In order to escape, he needs to figure out which corridors, if any, lead to an exit and freedom, rather than the certain doom of a pit or a ghost. The one sign of what lies behind the corridors is the wind: a pit produces a strong breeze (S) and an exit produces a weak breeze (W), while a ghost doesn't produce any breeze at all. Unfortunately, Pacman cannot measure the strength of the breeze at a specific corridor. Instead, he can stand between two adjacent corridors and feel the max of the two breezes. For example, if he stands between a pit and an exit he will sense a strong (S) breeze, while if he stands between an exit and a ghost, he will sense a weak (W) breeze. The measurements for all intersections are shown in the figure below. Also, while the total number of exits might be zero, one, or more, Pacman knows that two neighboring squares will not both be exits.

Pacman models this problem using variables X_i for each corridor *i* and domains P, G, and E.

1. State the binary and/or unary for this CSP (either implicitly or explicitly).
2. Cross out the values from the domains of the variables that will be deleted in enforcing arc consistency.



X_1	P	G	E
X_2	P	G	E
X_3	P	G	E
X_4	P	G	E
X_5	P	G	E
X_6	P	G	E

3. According to MRV, which variable or variables could the solver assign first?
4. Assume that Pacman knows that X_6 = G. List all the solutions of this CSP or write none if no solutions exist.

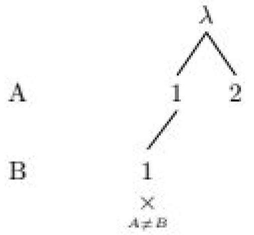
3 CSPs:

1. Consider the Constraint Satisfaction Problem (CSP) expressed as the constraint graph given in the lecture slides labeled at the bottom of the slides as D. Poole and A. Mackworth 2009 Artificial Intelligence Lecture 4.3, Page 7.
It has been made Arc Consistent on the slide labeled Lecture 4.3, Page 8.
The result of the first recursive descent into the variable elimination algorithm is given on the slide labeled Lecture 4.3, Page 9.
The resulting constraint graph is given on the slide labeled Lecture 4.3, Page 10.
(Additional introduction to **Generate and Test** Algorithm: Slide 10)
(a) How many leaf nodes are there in the Generate and Test tree for the original problem (Lecture 4.3, Page 7). Show how you computed this answer.

(b) How many leaf nodes are there in the Generate and Test tree for the arc consistent problem (Lecture 4.3, Page 8). Show how you computed this answer.

(c) The first iteration of the variable elimination algorithm has been provided. Show how the variable elimination algorithm gives the **solutions** for this CSP. Eliminate the variables in the following order: **D, E, B**. Your answer must show the **new relations generated** by the algorithm when **descending** in the recursion.

(d) With the constraint graph in part (b) above (the graph representing the CSP that has been preprocessed by AC-3), show how **backtracking search** can be used to solve this problem. To do this, you must draw the search tree generated to find all the answers. Each node in the tree represents a choice of a domain element for the variable represented by the level of the tree. Label the tree with a '×' underneath paths that can be cut off together with the constraint that caused the cutoff. The first such cutoff is shown below.



Use the following variable **ordering: A, B, D, E, and C**. Indicate (in a summary) the valid solution(s) that is (are) found.

Submission

Please submit your final answers in a **PDF** file through the OWL.

Please note you will not receive grades if TA can not recognize your handwriting, please finalize your solutions clearly and neatly and transform it to be a PDF version. Online tools for drawing diagrams like google drawing is recommended.

Part 2: Pacman Multi-Agent Search

Please check the Berkeley's cs188's Project 's page for detailed info and instruction.
Please download the code files A2-multiagent.zip.
You will essentially finish questions ***1, 2, 4*** in this project, except we have a different weighting: **Q1: 2%, Q2: 3%, Q4: 4% - total 9%.**

Submission

Please submit your final implementation python files: `multiAgents.py` through OWL.

Please *do not* change the other files in this distribution or submit any of our original files other than the expected files.