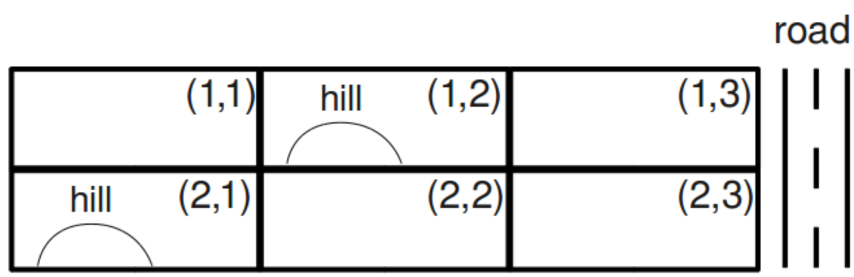
Written Assignment 2

Deadline: October 30th, 2019

Instruction: You may discuss these problems with classmates, but please complete the write-ups individually. (This applies to BOTH undergraduates and graduate students.) Remember the collaboration guidelines set forth in class: you may meet to discuss problems with classmates, but you may not take any written notes (or electronic notes, or photos, etc.) away from the meeting. Your answers must be **typewritten**, except for figures or diagrams, which may be hand-drawn. Please submit your answers (pdf format only) on **Canvas**.

Q1. Campus Layout (32 points, 4 points each question)

You are asked to determine the layout of a new, small college. The campus will have four structures: an administration structure (A), a bus stop (B), a classroom (C), and a dormitory (D). Each structure (including the bus stop) must be placed somewhere on the grid shown below.



The layout must satisfy the following constraints:

(i) The bus stop (B) must be adjacent to the road.

(ii) The administration structure (A) and the classroom (C) must both be adjacent to the bus stop (B).

(iii) The classroom (C) must be adjacent to the dormitory (D).

(iv) The administration structure (A) must not be adjacent to the dormitory (D).

(v) The administration structure (A) must not be on a hill.

(vi) The dormitory (D) must be on a hill or adjacent to the road.

(vii) All structures must be in different grid squares.

Here, adjacent means that the structures must share a grid edge, not just a corner.

Q1.1. Provide the domains of all variables after unary constraints have been applied.

Q1.2. Starting from the answer to Q1.1 (in which unary constraints are enforced), provide the domains of all variables after $A \to B$ is enforced.

Q1.3. Continuing from the answer to Q1.2, provide the domains of all variables after $C \to B$ is enforced.

Q1.4. What arcs got added to the queue while enforcing $C \to B$? Note that the queue contained $C \to D$, $D \to A$, $D \to B$, and $D \to C$ prior to enforcing $C \to B$.

Q1.5. Continuing from Q1.4, provide the domains of all variables after enforcing arc consistency until the queue is empty. Remember that the queue currently contains $C \to D$, $D \to A$, $D \to B$, and $D \to C$, and any arcs that were added while enforcing $C \to B$.

Q1.6. If arc consistency had resulted in all domains having a single value left, we would have already found a solution. Similarly, if it had found that any domain had no values left, we would have already found that no solution exists. Unfortunately, this is not the case in our example (as you should have found in the previous part). To solve the problem, we need to start searching. Use the MRV (minimum remaining values) heuristic to choose which variable gets assigned next (breaking any ties alphabetically). Which variable gets assigned next?

Q1.7. Continuing from Q1.6, the variable you selected should have two values left in its domain. We will use the least-constraining value (LCV) heuristic to decide which value to assign before continuing with the search. To choose which value is the least-constraining value, enforce arc consistency for each value (on a scratch piece of paper). For each value, count the total number of values remaining over all variables. Which value has the largest number of values remaining (and therefore is the least constraining value)?

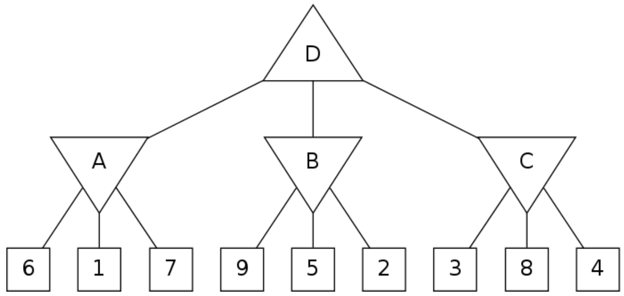
Q1.8. After assigning a variable, backtracking search with arc consistency enforces arc consistency before proceeding to the next variable.

Provide the domains of all variables after assignment of the least-constraining value to the variable you selected and enforcing arc consistency. Note that you already did this computation to determine which value was the LCV.

Q2. Minimax and Expectimax (20 points)

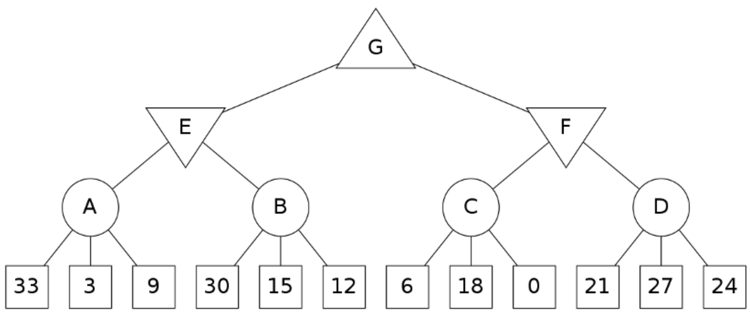
Q2.1. Minimax (10 points)

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. Outcome values for the maximizing player are listed for each leaf node, represented by the values in squares at the bottom of the tree. Assuming both players act optimally, carry out the minimax search algorithm. Enter the values for the letter nodes.



Q2.2. Expectimax (10 points)

Consider the game tree shown below. As in the previous problem, triangles that point up, such as the top node (root), represent choices for the maximizing player; triangles that point down represent choices for the minimizing player. The circular nodes represent chance nodes in which each of the possible actions may be taken with equal probability. The square nodes at the bottom represent leaf nodes. Assuming both players act optimally, carry out the expectiminimax search algorithm. Enter the values for the letter nodes.

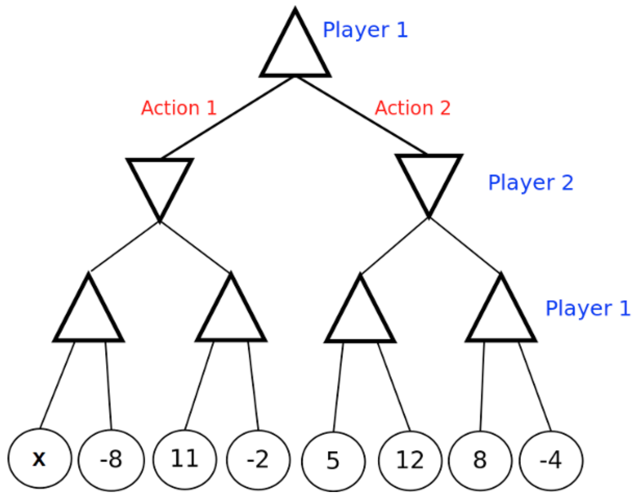


Q3. Unknown Leaf Value (28 points)

Consider the following game tree, where one of the leaves has an unknown payoff, x. Player 1 moves first, and attempts to maximize the value of the game. Each of the next 3 questions asks you to write a constraint on x specifying the set of values it can take. Please specify your answer in one of the following forms:

- Write All if x can take on all values
- Write None if x has no possible values
- Use an inequality in the form $x < \{value\}$, $x > \{value\}$, or $\{value1\} < x < \{value2\}$ to specify an interval of values. As an example, if you think x can take on all values larger than 16, you should enter $x > 16$.

Q3.1. Assume Player 2 is a minimizing agent (and Player 1 knows this). For what values of x is Player 1 guaranteed to choose Action 1 for their first move?



Q3.2. Assume Player 2 chooses actions at random with each action having equal probability (and Player 1 knows this). For what values of x is Player 1 guaranteed to choose Action 1?

Q3.3. Denote the minimax value of the tree as the value of the root when Player 1 is the maximizer and Player 2 is the minimizer. Denote the expectimax value of the tree as the value of the root when Player 1 is the maximizer and Player 2 chooses actions at random (with equal probability). For what values of x is the minimax value of the tree worth more than the expectimax value of the tree?

Q3.4. Is it possible to have a game, where the minimax value is strictly larger than the expectimax value?

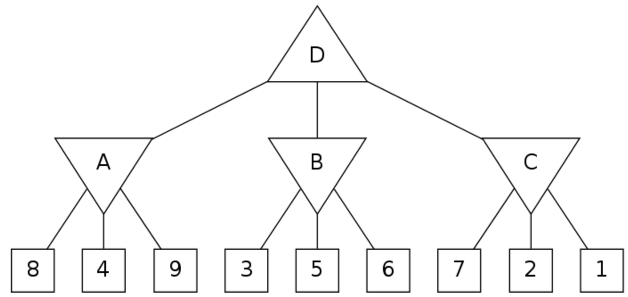
Q4. Alpha-Beta Pruning (20 points)

Consider the 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, use alpha-beta pruning to find the value of the root node. The search goes from left to right; when choosing which child to visit first, choose the left-most unvisited child.

Hint: Note that the value of a node where pruning occurs is not necessarily the maximum or minimum (depending on which node) of its children. When you prune on conditions, $V > \beta$ or $V < \alpha$, assume that the value of the node is V.

Q4.1. (10 points) Enter the values of the labeled nodes.

Q4.2. (10 points) Select the leaf nodes that don't get visited due to pruning.



Q5. Possible Pruning (grads only) (25 points)

Assume we run $\alpha - \beta$ pruning, expanding successors from left to right, on a game with tree as shown in figure below. Which of the following statements are true?

1. There exists an assignment of utilities to the terminal nodes such that no pruning will be achieved (shown in Figure (a)).
2. There exists an assignment of utilities to the terminal nodes such that the pruning shown in Figure (b) will be achieved.
3. There exists an assignment of utilities to the terminal nodes such that the pruning shown in Figure (c) will be achieved.
4. There exists an assignment of utilities to the terminal nodes such that the pruning shown in Figure (d) will be achieved.
5. There exists an assignment of utilities to the terminal nodes such that the pruning shown in Figure (e) will be achieved.
6. None of the above.