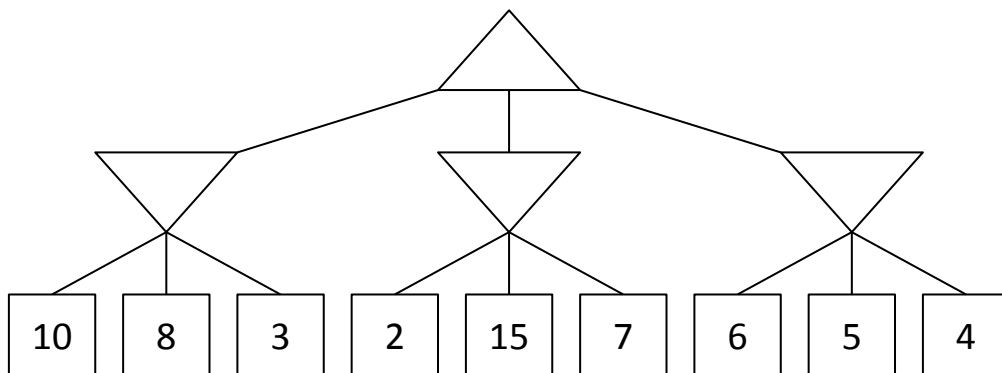


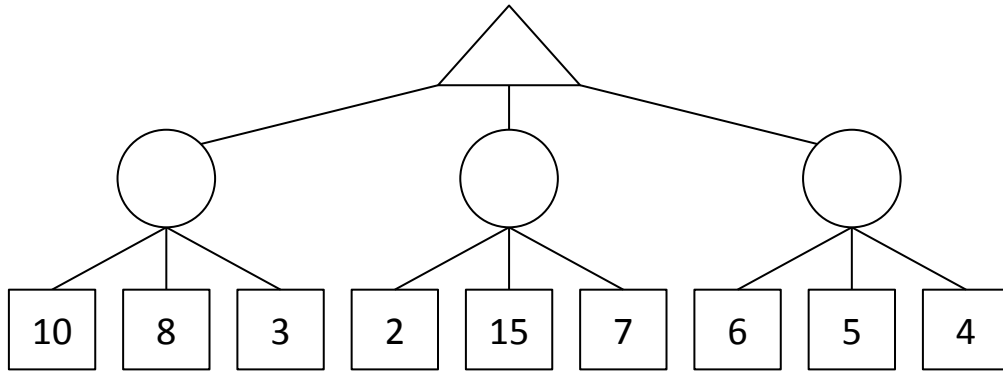
1 Games

- (a) 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.



- (b) 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.

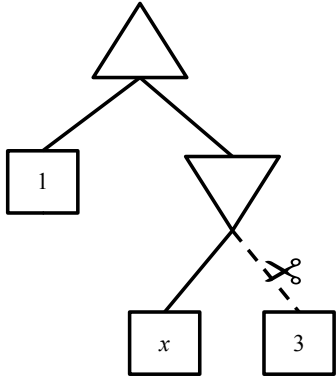
- (c) (optional) Again, consider the same zero-sum game tree, except that now, instead of a minimizing player, we have a chance node that will select one of the three values uniformly at random. Fill in the expectimax value of each node. The game tree is redrawn below for your convenience.



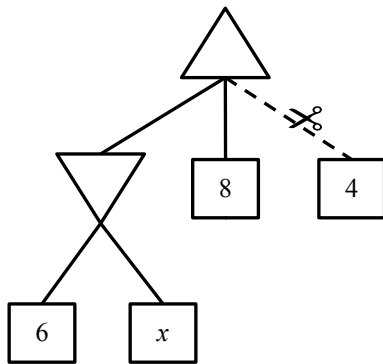
(d) (optional) Which nodes can be pruned from the game tree above through alpha-beta pruning? If no nodes can be pruned, explain why not.

Q2. Games: Alpha-Beta Pruning

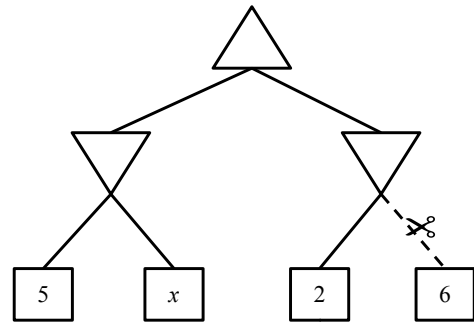
For each of the game-trees shown below, state for which values of x the dashed branch with the scissors will be pruned. If the pruning will not happen for any value of x write “none”. If pruning will happen for all values of x write “all”.



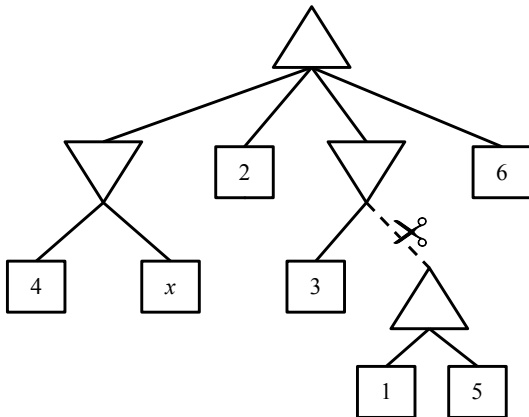
(a) Example Tree. Answer: $x \leq 1$.



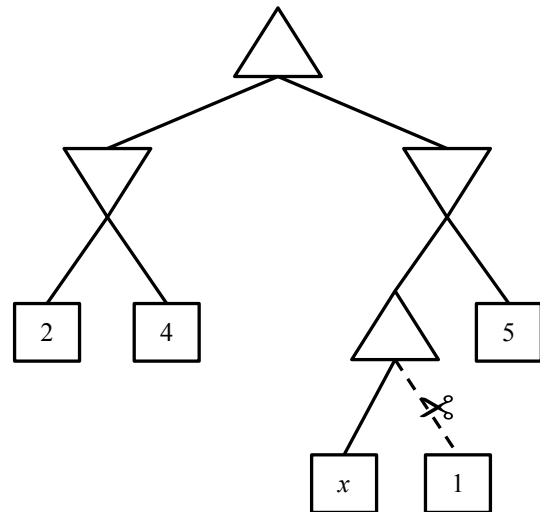
(b) Tree 1. Answer: _____



(c) Tree 2. Answer: _____



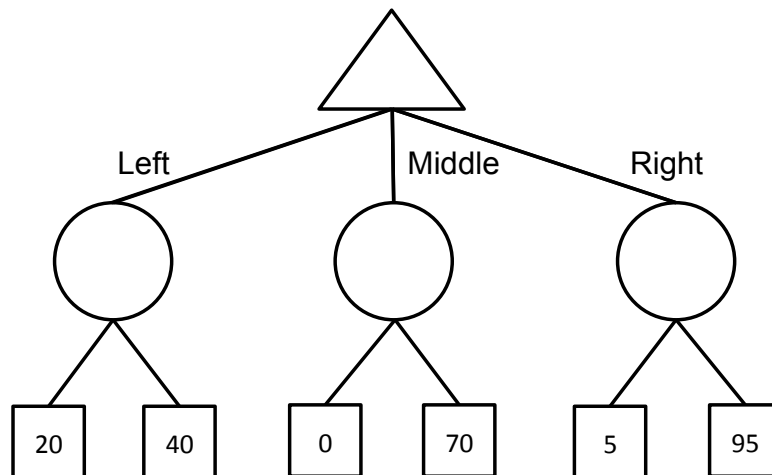
(d) Tree 3. Answer: _____



(e) Tree 4. Answer: _____

Q3. Bounded Expectimax

- (a) **Expectimax.** Consider the game tree below, where the terminal values are the *payoffs* of the game. Fill in the expectimax values, assuming that player 1 is maximizing expected payoff and player 2 plays uniformly at random (i.e., each action available has equal probability).



- (b) Again, assume that Player 1 follows an expectimax strategy (i.e., maximizes expected payoff) and Player 2 plays uniformly at random (i.e., each action available has equal probability).

(i) What is Player 1's expected payoff if she takes the expectimax optimal action?

(ii) Multiple outcomes are possible from Player 1's expectimax play. What is the worst possible payoff she could see from that action?

- (c) Even if the average outcome is good, Player 1 doesn't like that very bad outcomes are possible. Therefore, rather than purely maximizing expected payoff using expectimax, Player 1 chooses to perform a modified search. In particular, she only considers actions whose worst-case outcome is 10 or better.

(i) Which action does Player 1 choose for this tree?

(ii) What is the expected payoff for that action?

(iii) What is the worst payoff possible for that action?

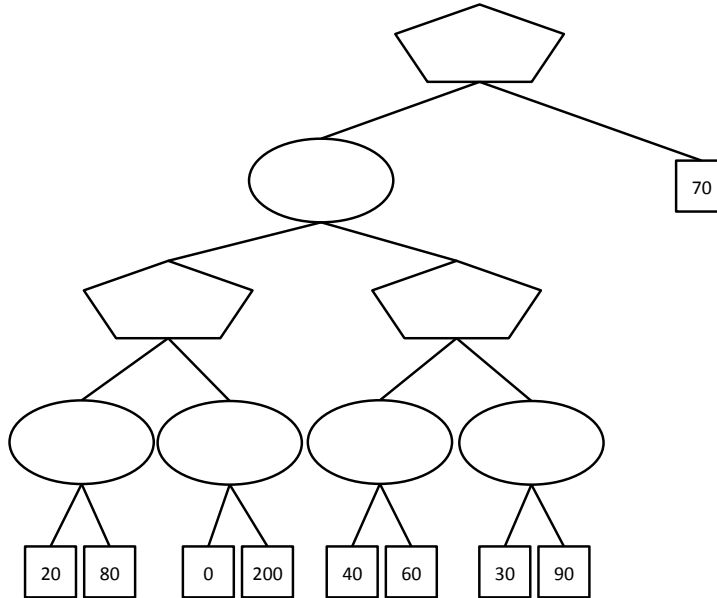
- (d) Now let's consider a more general case. Player 1 has the following preferences:

- Player 1 prefers any lottery with worst-case outcome of 10 or higher over any lottery with worst-case outcome lower than 10.

- Among two lotteries with worst-case outcome of 10 or higher, Player 1 chooses the one with the highest expected payoff.
- Among two lotteries with worst-case outcome lower than 10, Player 1 chooses the one with the highest worst-case outcome (breaking ties by highest expected payoff).

Player 2 still always plays uniformly at random.

To compute the appropriate values of tree nodes, Player 1 must consider both expectations and worst-case values at each node. For each node in the game tree below, fill in a pair of numbers (e, w) . Here e is the expected value under Player 1's preferences and w is the value of the worst-case outcome under those preferences, assuming that Player 1 and Player 2 play according to the criteria described above.



(e) Now let's consider the general case, where the lower bound used by Player 1 is a number L not necessarily equal to 10, and not referring to the particular tree above. Player 2 still plays uniformly at random.

(i) Suppose a Player 1 node has two children: the first child passes up values (e_1, w_1) , and the second child passes up values (e_2, w_2) . What values (e, w) will be passed up by a Player 1 node if

1. $w_1 < w_2 < L$
2. $w_1 < L < w_2$
3. $L < w_1 < w_2$

(ii) Now consider a Player 2 node with two children: the first child passes up values (e_1, w_1) and the second child passes up values (e_2, w_2) . What values (e, w) will be passed up by a Player 2 node if

1. $w_1 < w_2 < L$
2. $w_1 < L < w_2$
3. $L < w_1 < w_2$