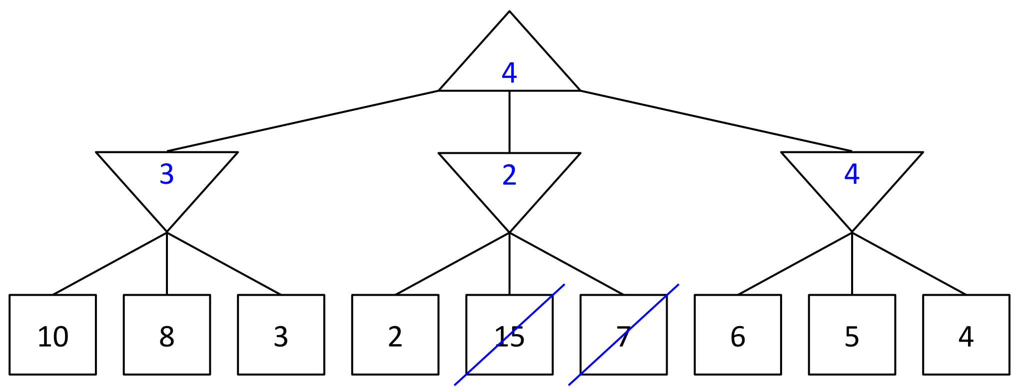
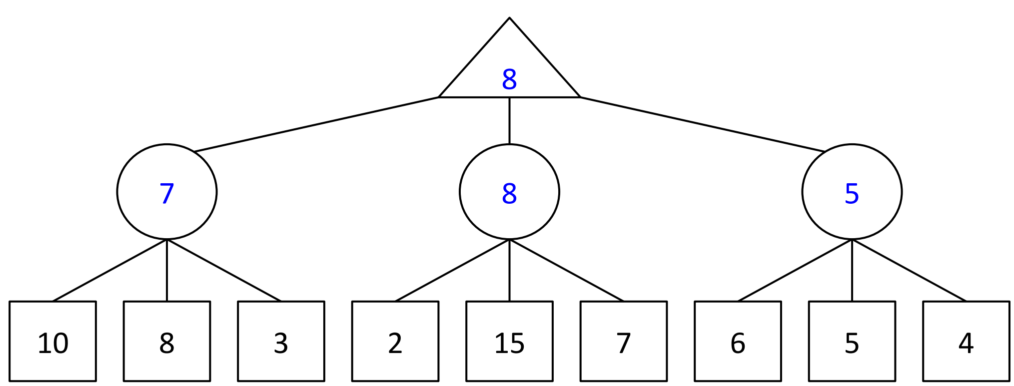
1 Games

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.

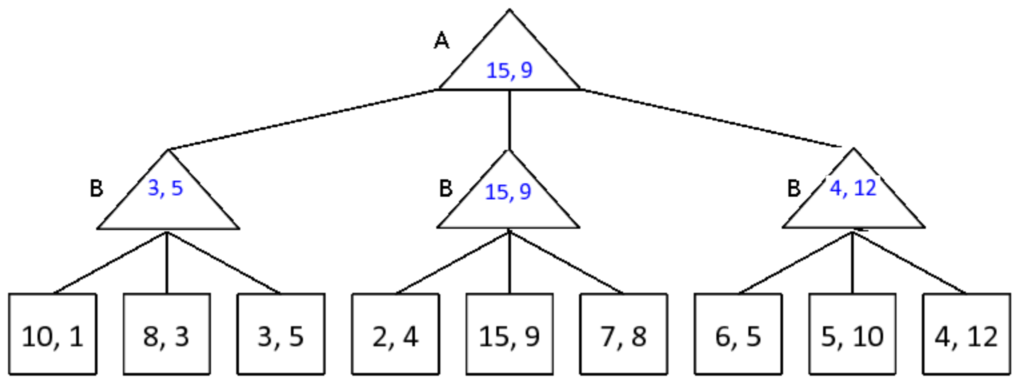
3. (optional) Again, consider the same zero-sum game tree, except that now, instead of a minimizing player, we have a chance node that will select one of the three values uniformly at random. Fill in the expectimax value of each node. The game tree is redrawn below for your convenience.



4. (optional) Which nodes can be pruned from the game tree above through alpha-beta pruning? If no nodes can be pruned, explain why not. No nodes can be pruned. There will always be the possibility that an as-yet-unvisited leaf of the current parent chance node will have a very high value, which increases the overall average value for that chance node. For example, when we see that leaf 4 has a value of 2, which is much less than the value of the left chance node, 7, at this point we cannot make any assumptions about how the value of the middle chance node will ultimately be more or less in value than the left chance node. As it turns out, the leaf 5 has a value of 15, which brings the expected value of the middle chance node to 8, which is greater than the value of the left chance node. In the case where there is an upper bound to the value of a leaf node, there is a possibility of pruning: suppose that an upper bound of +10 applies only to the children of the rightmost chance node. In this case, after seeing that leaf 7 has a value of 6 and leaf 8 has a value of 5, the best possible value that the rightmost chance node can take on is $\frac{6+5+10}{3} = 7$, which is less than 8, the value of the middle chance node. Therefore, it is possible to prune leaf 9 in this case.

2 (Optional) Nonzero-sum Games

1. Let's look at a non-zero-sum version of a game. In this formulation, player A's utility will be represented as the first of the two leaf numbers, and player B's utility will be represented as the second of the two leaf numbers. Fill in this non-zero game tree assuming each player is acting optimally.



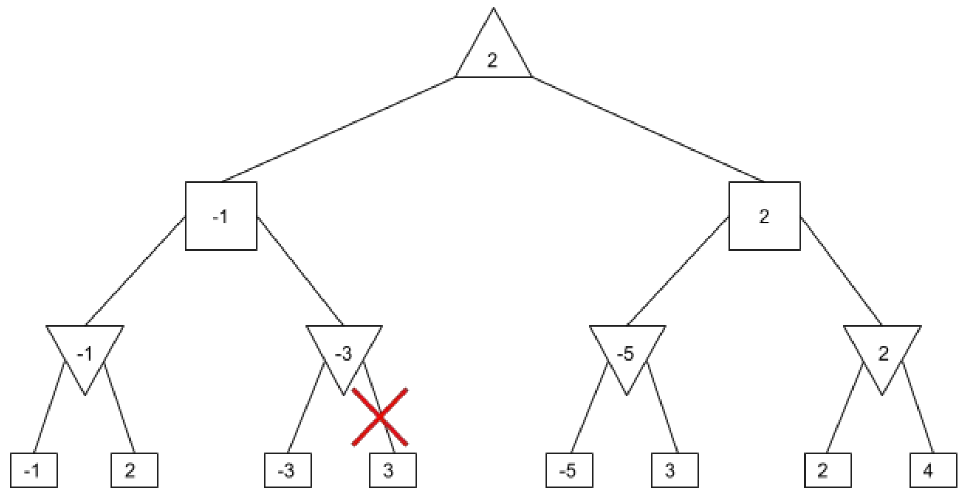
2. Which nodes can be pruned from the game tree above through alpha-beta pruning? If no nodes can be pruned, explain why not. No nodes can be pruned. Because this game is non-zero-sum, there can exist a leaf node anywhere in the tree that is good for both player A and player B.

Q3. Fair Play

Consider a game tree with three agents: a maximizer, a minimizer, and an equalizer. The maximizer chooses the highest score, the minimizer chooses the lowest score, and the equalizer tries to *minimize the absolute value* (i.e. equalizer wants to make the game as close as possible, so it chooses whichever value is closest to zero).

We use an upward-facing triangle to represent a max node, a downward-facing triangle to represent a min node, and a square to represent an equalizer node. The values in the leaves are given from max's point of view.

(a) Fill in all values in the game tree below:

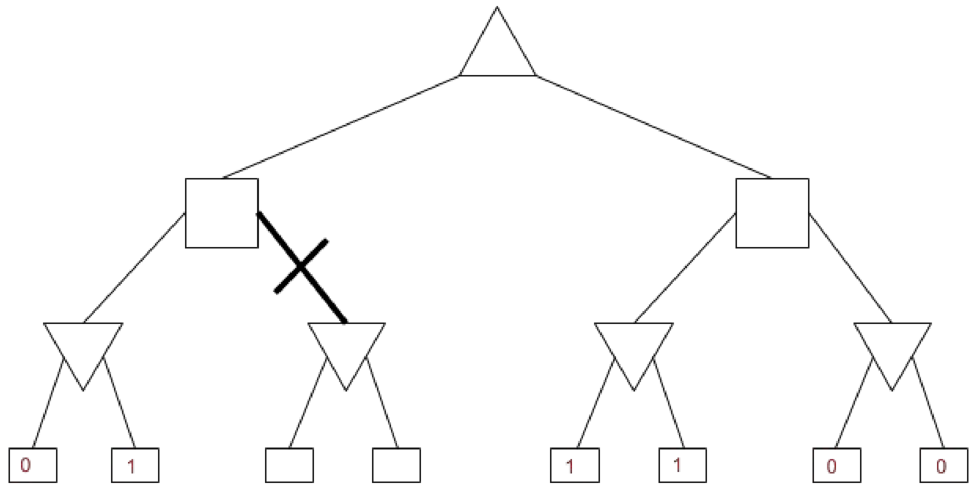


(b) In the same game tree above, put an X on the line of all branches that can be pruned, or write "No pruning possible." Assume that branches are explored from left to right.

We can prune this node because once min sees a -3, the min node is worth -3 or lower, so equalizer prefers -1, and the fourth leaf is irrelevant. nNote that we cannot prune the sixth leaf: even though the min node is -5 or worse, suppose the seventh and eighth leaves were +10; in that case, if the sixth leaf were -20, the equalizer would prefer the +10 to the -20, causing max to prefer moving right, whereas if in the same case the sixth leaf were -5, the equalizer would prefer the -5 to the +10, causing max to prefer moving left. Hence in some situations the value of the sixth leaf matters. (Note that in min/max trees, pruning can be decided *without* considering the values that subsequent leaves might have.) This reasoning is tricky, so we took off only 1 point for pruning the sixth leaf.

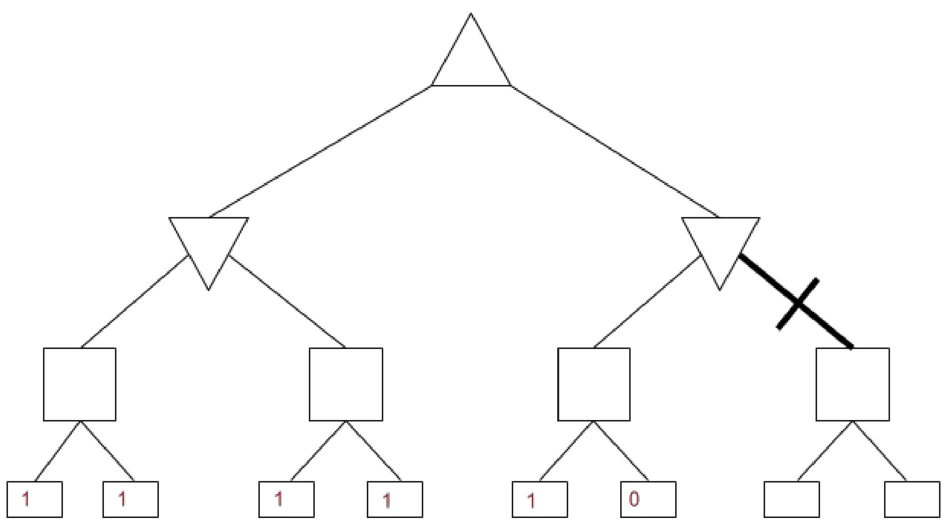
(c) For each of the following game trees, fill in values in the leaf nodes such that only the marked, bold branches can be pruned. Assume that branches are explored from left to right. If no values will allow the indicated nodes to be pruned, write "Not possible." **Be very clear**: if you write "Not possible," we will not look at the values you filled in.

(i) [Hint: what is the *best possible value* from the equalizer's viewpoint?]



0 is the optimal value for the equalizer, so the marked subtree is irrelevant. Note that you also have to put in values for the other leaves so that *no* pruning takes place!

(ii) Note that the order of the players has changed in the game tree below.



Here the pruning is essentially standard alpha-beta pruning because the top two levels of the tree are max and min.