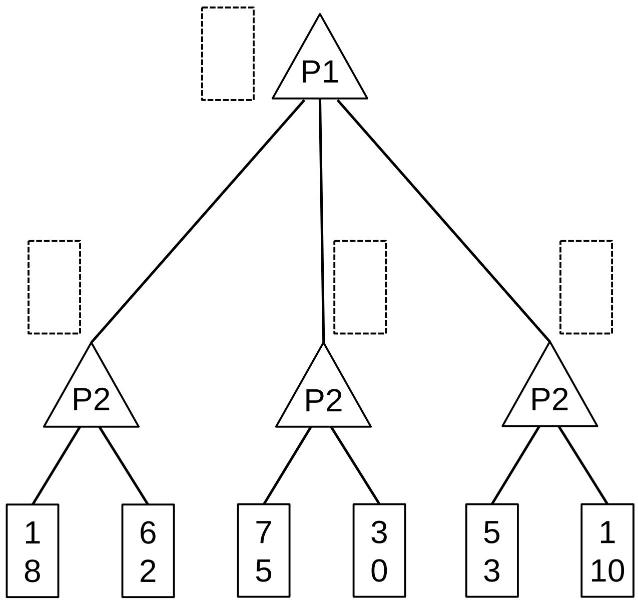
Midterm Review Games Solutions

Q1. Games

For the following game tree, each player maximizes their respective utility. Let x, y respectively denote the top and bottom values in a node. Player 1 uses the utility function $U_1(x, y) = x$.



(a) Both players know that Player 2 uses the utility function $U_2(x, y) = x - y$.

(i) Fill in the rectangles in the figure above with pair of values returned by each max node. From top-down, left-right: $(6, 2), (6, 2), (3, 0), (5, 3)$

(ii) You want to save computation time by using pruning in your game tree search. On the game tree above, put an 'X' on branches that do not need to be explored or simply write 'None'. Assume that branches are explored from left to right. None.

(b) Now assume Player 2 changes their utility function based on their mood. The probabilities of Player 2's utilities and mood are described in the following table. Let M, U respectively denote the mood and utility function of Player 2.

$P(M = happy)$	$P(M = mad)$
a	b

	$M = happy$	$M = mad$
$P(U_2(x, y) = -x \mid M)$	c	f
$P(U_2(x, y) = x - y \mid M)$	d	g
$P(U_2(x, y) = x^2 + y^2 \mid M)$	e	h

(i) Calculate the maximum expected utility of the game for Player 1 in terms of the values in the game tree and the tables. It may be useful to record and label your intermediate calculations. You may write your answer in terms of a max function.
We first calculate the new probabilities of each utility function as follows.

$P(U_2(x,y) = -x)$	$P(U_2(x,y) = x - y)$	$P(U_2(x,y) = x^2 + y^2)$
$ac + bf$	$ad + bg$	$ae + bh$

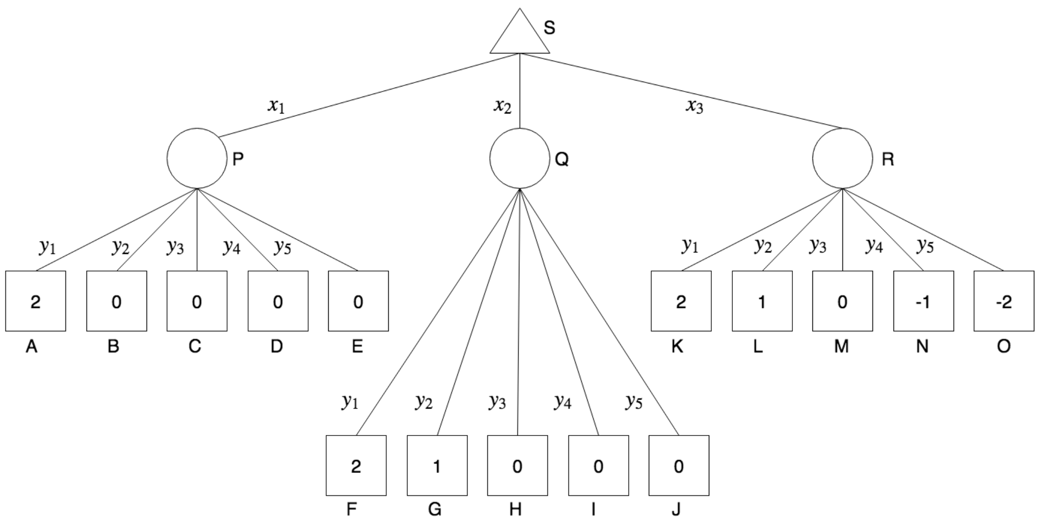
$EU(\text{Left Branch}) = (ac + bf)(1) + (ad + bg)(6) + (ae + bh)(1)$
$EU(\text{Middle Branch}) = (ac + bf)(3) + (ad + bg)(3) + (ae + bh)(7)$
$EU(\text{Right Branch}) = (ac + bf)(1) + (ad + bg)(5) + (ae + bh)(1)$

$MEU(\phi) = max((ac + bf)(1) + (ad + bg)(6) + (ae + bh)(1), (ac + bf)(3) + (ad + bg)(3) + (ae + bh)(7), (ac + bf)(1) + (ad + bg)(5) + (ae + bh)(1))$

Q2. Bike Bidding Battle

Alyssa P. Hacker and Ben Bitdiddle are bidding in an auction at Stanley University for a bike. Alyssa will either bid x_1, x_2, or x_3 for the bike. She knows that Ben will bid y_1, y_2, y_3, y_4, or y_5, but she does not know which. All bids are nonnegative.

(a) Alyssa wants to maximize her payoff given by the expectimax tree below. The leaf nodes show Alyssa's payoff. The nodes are labeled by letters, and the edges are labeled by the bid values x_i and y_i. The maximization node S represents Alyssa, and the branches below it represent each of her bids: x_1, x_2, x_3. The chance nodes P, Q, R represent Ben, and the branches below them represent each of his bids: y_1, y_2, y_3, y_4, y_5.



(i) Suppose that Alyssa believes that Ben would bid any bid with equal probability. What are the values of the chance (circle) and maximization (triangle) nodes?

Node P: 0.4 Node Q: 0.6 Node R: 0 Node S: 0.6

(ii) Based on the information from the above tree, how much should Alyssa bid for the bike?

○ x_1 ● x_2 ○ x_3

(b) Alyssa does expectimax search by visiting child nodes from left to right. Ordinarily expectimax trees cannot be pruned without some additional information about the tree. Suppose, however, that Alyssa knows that the leaf nodes are ordered such that payoffs are non-increasing from left to right (the leaf nodes of the above diagram is an example of this ordering). Recall that if node X is a child of a maximizer node, a child of node X may be pruned if we know that the value of node X will never be $>$ some threshold (in other words, it is $\leq$ that threshold). Given this information, if it is possible to prune any branches from the tree, mark them below. Otherwise, mark "None of the above."

☐ A ☐ B ☐ C ☐ D ☐ E ☐ F ☐ G ☐ H
☐ I ☐ J ☐ K ☐ L ☐ M ■ N ■ O ○ None of the above

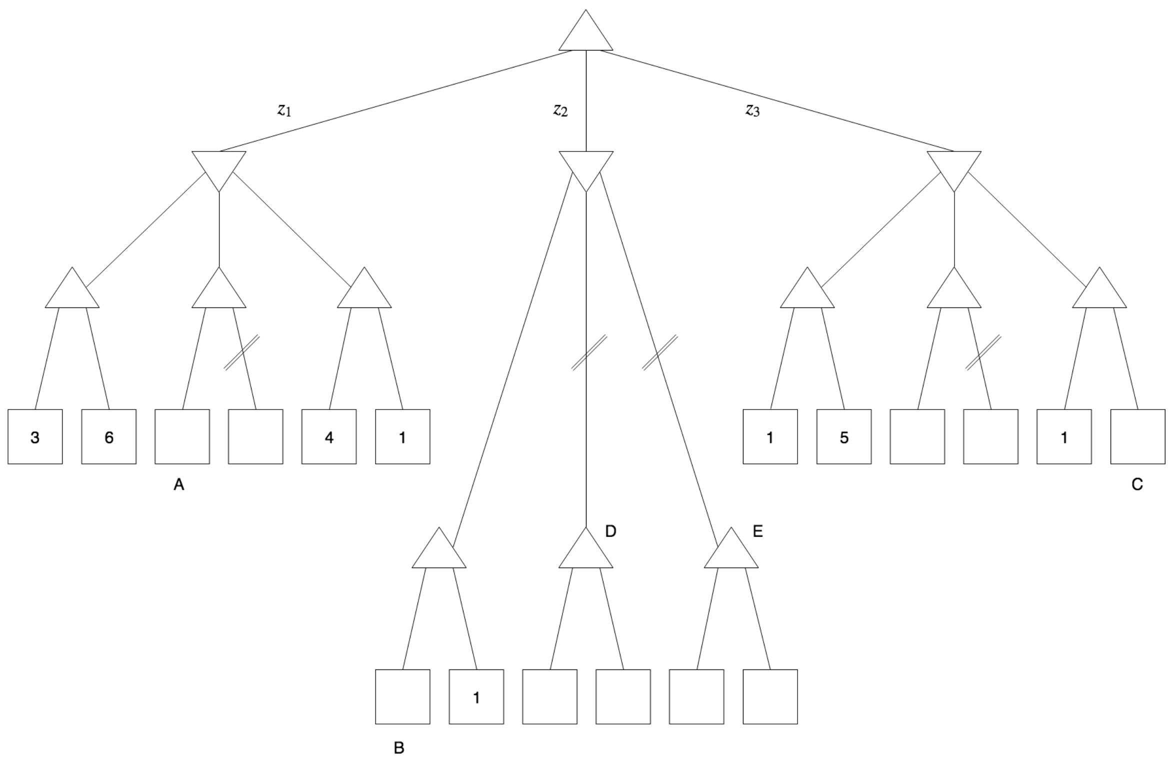
To prune the children of a chance node in an expectimax tree, Alyssa would need to keep track of a threshold on the value of the chance node: if at some point while searching left to right, she realizes that the value of the chance node will never be higher than its threshold, she can prune the remaining children of the chance node.

Alyssa needs to search the entire left subtree because she does not have a threshold against which to compare the value of P.

When Alyssa searches the center subtree, she knows that the maximizing node will only consider taking action x_2 if the value of Q is higher than the value of P, which is 0.4. If at some point Alyssa realizes that the value of Q will never be higher than 0.4, she can prune the remaining children. After exploring node G, Alyssa knows that nodes H, I, and J are ≤ 1, which means that the value of node Q is at most 1.2. After exploring node H, Alyssa knows that nodes I and J are ≤ 0, which means that the value of node Q is at most 0.6. After exploring node I, Alyssa knows that node J is ≤ 0, which means that the value of node Q is at most 0.6. This is not enough information to prune any of the nodes in the center subtree because at no point does Alyssa know for sure that the value of Q is ≤ 0.4.

When Alyssa searches the right subtree, if at some point Alyssa ralizes that the value of R will never be higher than 0.6, then she can prune the remaining of children of R. After exploring node L, Alyssa knows that the nodes M, N, and O are ≤ 1, which means that the value of node R is at most 1.2. After exploring node M, Alyssa knows that nodes N and O are ≤ 0, which means that the value of node Q is at most 0.6. At this point, Alyssa can prune nodes N and O because they can only make the value of R lower than the value of Q.

(c) Unrelated to parts (a) and (b), consider the minimax tree below. whose leaves represent payoffs for the maximizer. The crossed out edges show the edges that are pruned when doing naive alpha-beta pruning visiting children nodes from left to right. Assume that we prune on equalities (as in, we prune the rest of the children if the current child is $\leq \alpha$ (if the parent is a minimizer) or $\geq \beta$ (if the parent is a maximizer)).



Fill in the inequality expressions for the values of the labeled nodes A and B. Write ∞ and $-\infty$ if there is no upper or lower bound, respectively.

1. $\boxed{6} \leq \text{A} \leq \boxed{\infty}$

2. $\boxed{-\infty} \leq \text{B} \leq \boxed{4}$

(d) Suppose node B took on the largest value it could possibly take on and still be consistent with the pruning scheme above. After running the pruning algorithm, we find that the values of the left and center subtrees have the same minimax value, both 1 greater than the minimax value of the right subtree. Based on this information, what is the numerical value of node C?

○ 1 ○ 2 ● 3 ○ 4 ○ 5 ○ 6 ○ 7 ○ 8 ○ 9 ○ 10

(e) For which values of nodes D and E would choosing to take action z_2 be *guaranteed* to yield the same payoff as action z_1? Write ∞ and $-\infty$ if there is no upper or lower bound, respectively (this would correspond to the case where nodes D and E can be any value).

1. $\boxed{4} \leq \text{D} \leq \boxed{\infty}$

2. $\boxed{4} \leq \text{E} \leq \boxed{\infty}$

When doing naive alpha-beta pruning, the values propagated up to the parent nodes are not necessarily exact, but rather bounds. If $D < 4$ or $E < 4$, then the true minimax value of the center subtree is less than the true minimax value of the left subtree.