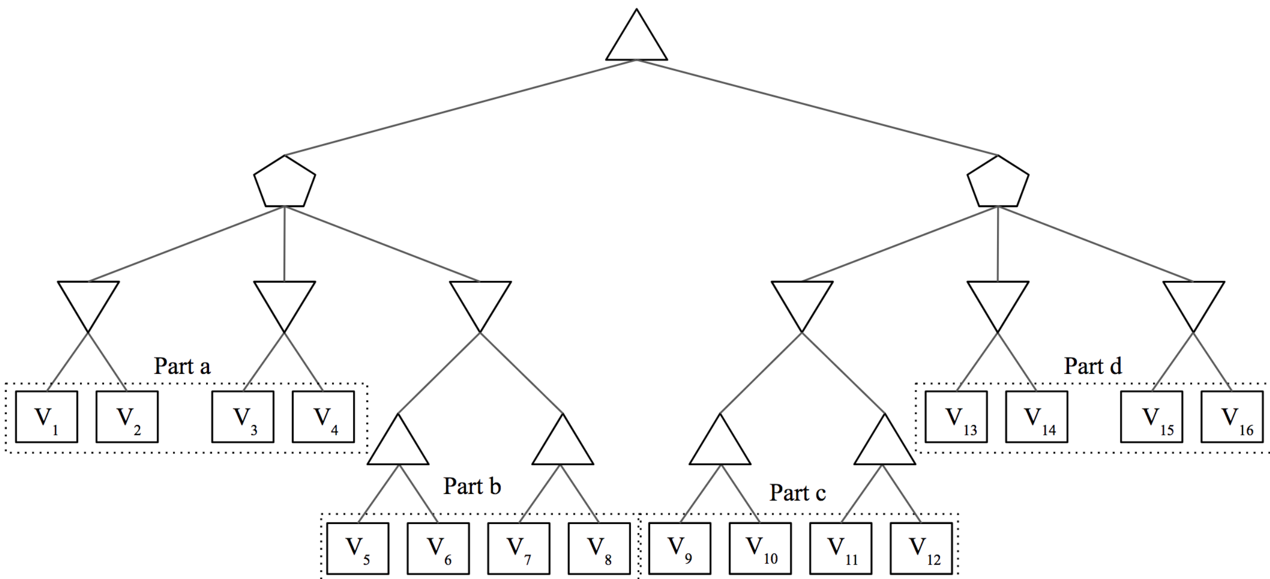
CS 188
Summer 2023

Discussion 2C Solutions

Q1. MedianMiniMax

You're living in utopia! Despite living in utopia, you still believe that you need to maximize your utility in life, other people want to minimize your utility, and the world is a 0 sum game. But because you live in utopia, a benevolent social planner occasionally steps in and chooses an option that is a compromise. Essentially, the social planner (represented as the pentagon) is a median node that chooses the successor with median utility. Your struggle with your fellow citizens can be modelled as follows:



There are some nodes that we are sometimes able to prune. In each part, mark all of the terminal nodes such that **there exists a possible situation** for which the node **can be pruned**. In other words, you must consider **all** possible pruning situations. Assume that evaluation order is **left to right** and all V_i's are **distinct.**

Note that as long as there exists ANY pruning situation (does not have to be the same situation for every node), you should mark the node as prunable. Also, alpha-beta pruning does not apply here, simply prune a sub-tree when you can reason that its value will not affect your final utility.

(a)
- □ V_1
- □ V_2
- □ V_3
- □ V_4
- ■ None

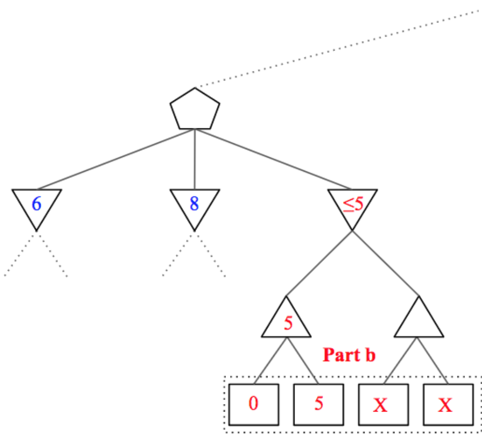
(b)
- □ V_5
- ■ V_6
- ■ V_7
- ■ V_8
- □ None

(c)
- □ V_9
- □ V_{10}
- ■ V_{11}
- ■ V_{12}
- □ None

(d)
- □ V_{13}
- ■ V_{14}
- ■ V_{15}
- ■ V_{16}
- □ None

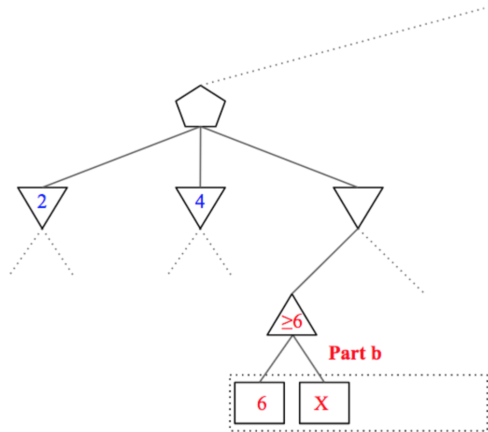
Part a:

For the left median node with three children, at least two of the childrens' values must be known since one of them will be guaranteed to be the value of the median node passed up to the final maximizer. For this reason, none of the nodes in part a can be pruned.



The value of this subtree will only get smaller.

The median node will **NOT** choose the value of this subtree. 6 is the median.



The value of this subtree will only get bigger.

*If the value of this subtree is chosen by the minimizer**, it will **NOT** be chosen by the median node.

**It is possible that the median is the value of the subtree to the right that we haven't looked at yet*

Part b (pruning V_7, V_8):

Let min_1, min_2, min_3 be the values of the three minimizer nodes in this subtree.

In this case, we may not need to know the final value min_3. The reason for this is that we may be able to put a bound on its value after exploring only partially, and determine the value of the median node as either min_1 or min_2 if $min_3 \leq \min(min_1, min_2)$ or $min_3 \geq \max(min_1, min_2)$.

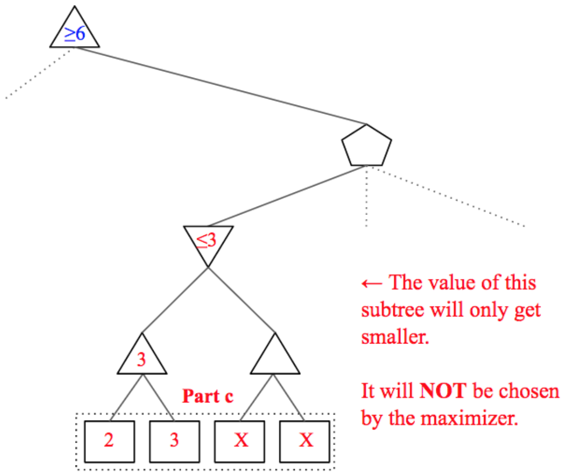
We can put an upper bound on min_3 by exploring the left subtree V_5, V_6 and if $\max(V_5, V_6)$ is lower than both min_1 and min_2, the median node's value is set as the smaller of min_1, min_2 and we don't have to explore V_7, V_8 in Figure 1.

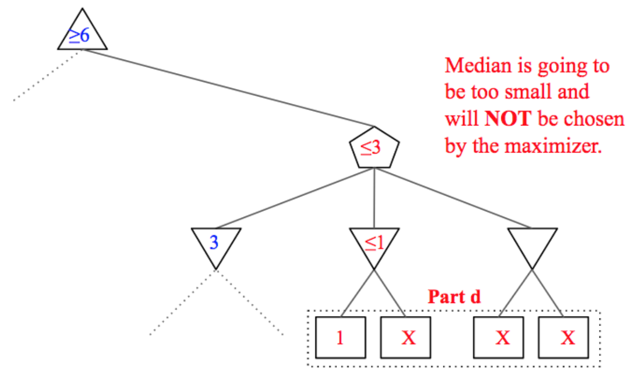
Part b (pruning V_6):

It's possible for us to put a lower bound on min_3. If V_5 is larger than both min_1 and min_2, we do not need to explore V_6.

The reason for this is subtle, but if the minimizer chooses the left subtree, we know that $min_3 \geq V_5 \geq \max(min_1, min_2)$ and we don't need V_6 to get the correct value for the median node which will be the larger of min_1, min_2.

If the minimizer chooses the value of the right subtree, the value at V_6 is unnecessary again since the minimizer never chose its subtree.





Part c (pruning V_{11}, V_{12}):
Assume the highest maximizer node has a current value $max_1 \geq Z$ set by the left subtree and the three minimizers on this right subtree have value min_1, min_2, min_3.

In this part, if $min_1 \leq \max(V_9, V_{10}) \leq Z$, we do not have to explore V_{11}, V_{12}. Once again, the reasoning is subtle, but we can now realize if either $min_2 \leq Z$ or $min_3 \leq Z$ then the value of the right median node is for sure $\leq Z$ and is useless.

Only if both $min_2, min_3 \geq Z$ will the whole right subtree have an effect on the highest maximizer, but in this case the exact value of min_1 is not needed, just the information that it is $\leq Z$. Clearly in both cases, V_{11}, V_{12} are not needed since an exact value of min_1 is not needed.

We will also take the time to note that if $V_9 \geq Z$ we do have to continue the exploring as V_{10} could be even greater and the final value of the top maximizer, so V_{10} can't really be pruned.

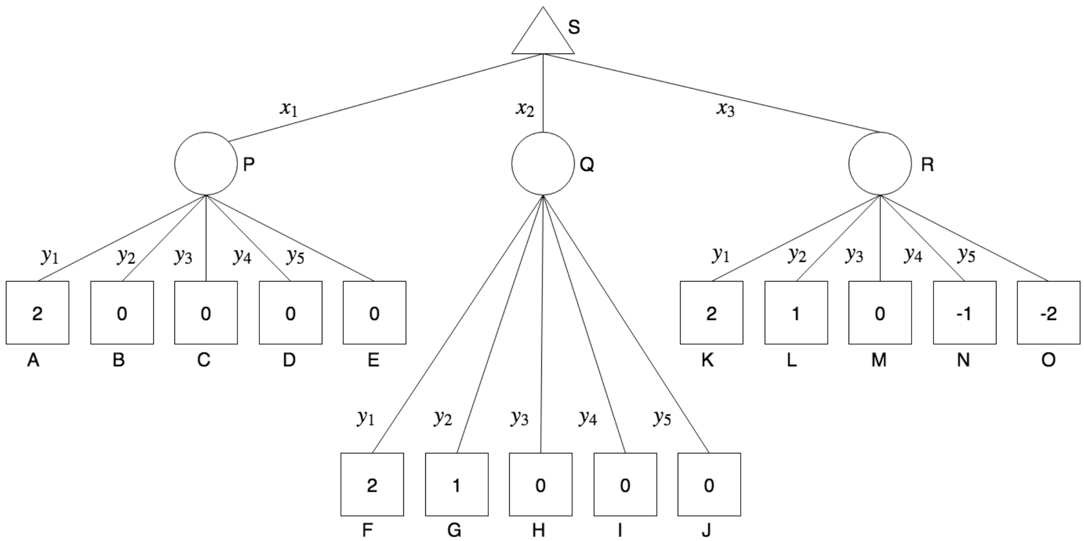
Part d (pruning V_{14}, V_{15}, V_{16}):
Continuing from part c, if we find that $min_1 \leq Z$ and $min_2 \leq Z$ we can stop.

We can realize this as soon we explore V_{13}. Once we figure this out, we know that our median node's value must be one of these two values, and neither will replace Z so we can stop.

Q2. [Optional] 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

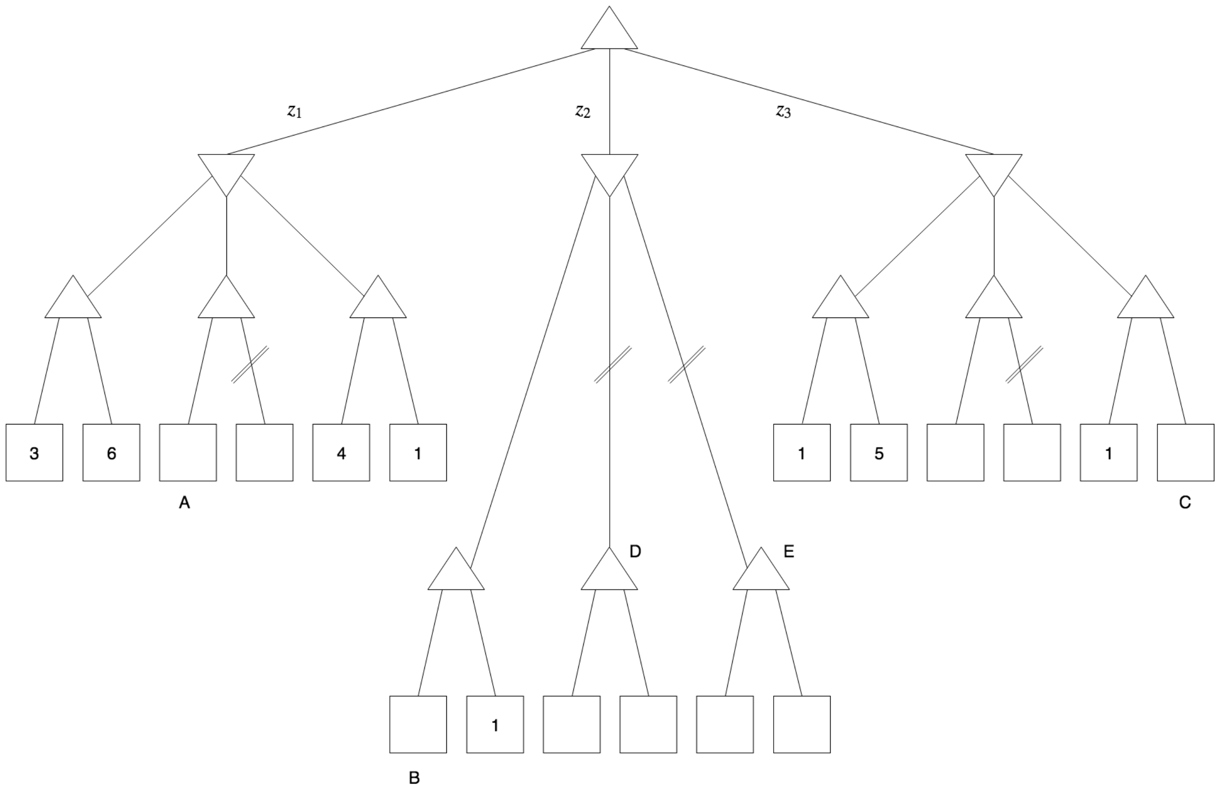
To prune the children of a chance node in an expectimax tree, Alyssa would need to keep track of a threshold on the value of the chance node: if at some point while searching left to right, she realizes that the value of the chance node will never be higher than its threshold, she can prune the remaining 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. $6 \leq A \leq \infty$

2. $-\infty \leq B \leq 4$

(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. $4 \leq D \leq \infty$

2. $4 \leq E \leq \infty$

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.