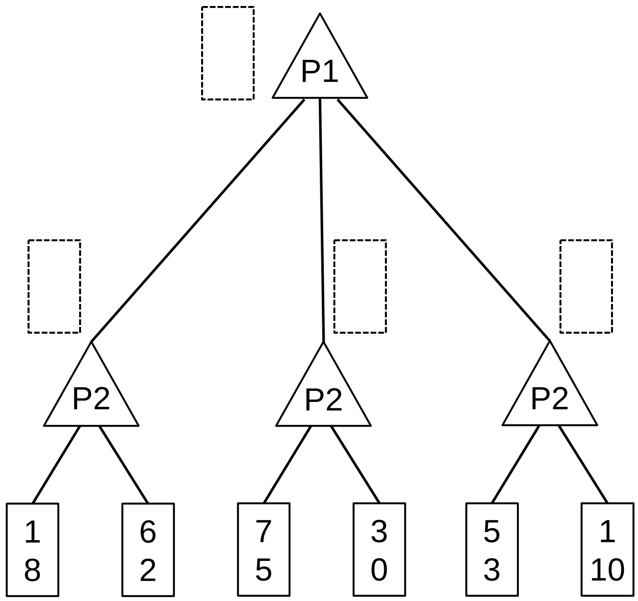
CS 188 Summer 2023 Midterm Review Games Solutions

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.

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

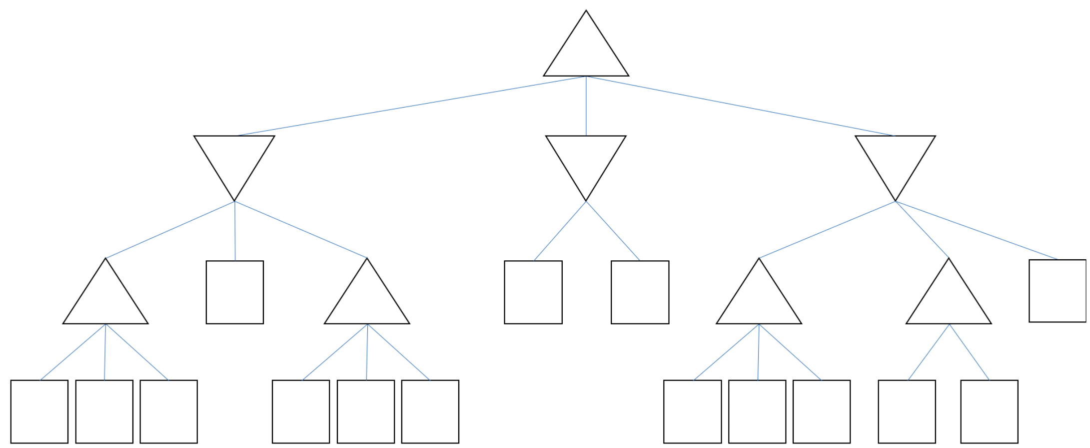
$EU(\text{Left Branch}) = (ac + bf)(1) + (ad + bg)(6) + (ae + bh)(1)$
$EU(\text{Middle Branch}) = (ac + bf)(3) + (ad + bg)(3) + (ae + bh)(7)$
$EU(\text{Right Branch}) = (ac + bf)(1) + (ad + bg)(5) + (ae + bh)(1)$

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

Q2. Tom and Jerry

Bored at home, Tom and Jerry decided to play a game.
For this question, assume that branches are visited in left to right order.

(a) To analyze the game, Tom drew a game tree. However, Jerry quickly erased all the leaf node values in the tree, as shown below, and asked Tom to think about running alpha-beta pruning on game trees with the same structure.

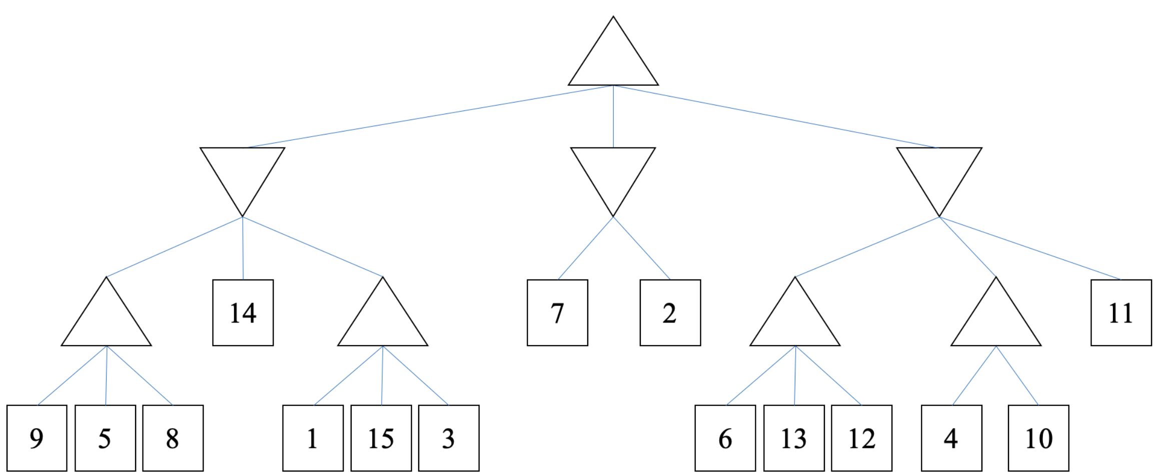


(i) The maximum possible number of leaf nodes pruned = 6

(ii) The minimum possible number of leaf nodes pruned = 0 .

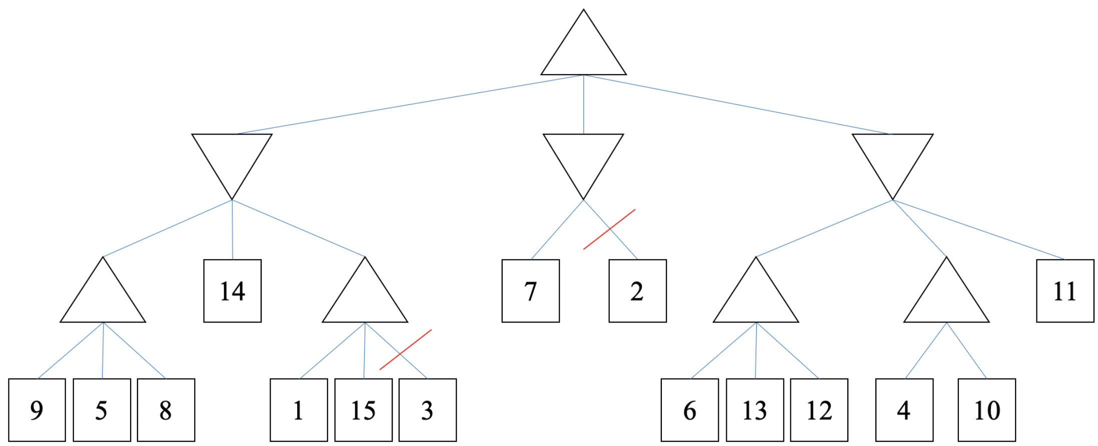
For the explanation, see part c.

Tom answered the questions correctly, so Jerry filled out the leaf node values.

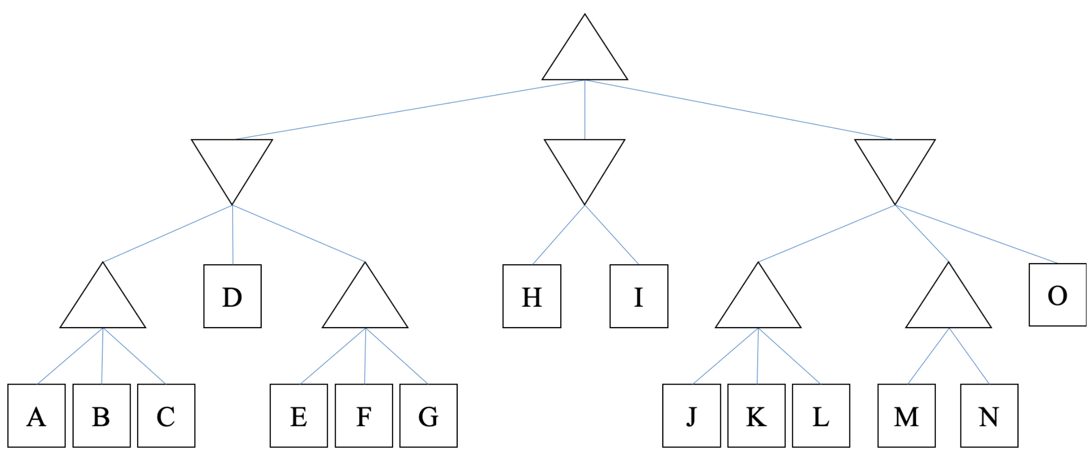


(b) Using alpha beta pruning, how many leaf nodes can be pruned?

2 .



(c) Tom rearranges the **leaf node values from the tree above** such that alpha beta pruning prunes the maximum amount of leaf nodes. Each leaf node value could have been moved to any of the other leaf nodes (or not moved at all). Assume we are talking about such a tree for all subquestions in this part. **Note, there are 15 leaf nodes and 15 possible values from 1-15 so each value is used exactly once.**
Hint: Tom has a few options in how to rearrange leaves to get max pruning. Try to derive constraints on the values that need to hold for max pruning to happen.
Use this tree with place holder letters to answer the questions below.



The maximal pruning can have F, G, I, M, N and O pruned.
To get I pruned, H must be smaller than the left minimizer, which is also the root value.
To get F and G pruned, E must be larger than the root value, which is the smaller one between D or max(A, B, C).
To get M, N and O pruned, max(J, K and L) must be smaller than the root value, which means J, K and L are all smaller than the root value.

The largest result is therefore 13, where we have max(A, B, C), D, and max(E, F, G) each taking a value from 15, 14 and 13.

The smallest result is 5. We've shown that H, J, K and L are smaller than the root value. To make the root smallest, we should let D be smaller than A, B, and C. In this case, root = D, and is larger than H, J, K and L, so root = 5. If root = max(A, B, C), it is at least 7.

(i) If the value of the root is 8, which of the following leaf nodes are guaranteed to have value < 8?

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

(ii) If the value of the root is 8, which of the following leaf nodes are guaranteed to have value ≥ 8?

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

(iii) If the value of the root is 8, which of the following leaf nodes are guaranteed to have value 8?

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

Since the root can come from max(A,B,C) or D neither can be guaranteed to be 8.

(iv) If the value of the root is 6, which of the following leaf nodes are guaranteed to have value 6?

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

There are 2 possibilities for the root: it is max(A, B, C), or it is D. If it is max(A, B, C), the root is at least 7. So if the root is 8, it could be either max(A, B, C) or D, so we cannot guarantee any of these leaf node values.

If the root is 6, it must be D.

(v) Which of the results (root values) are possible for Tom's rearranged game tree?

☐ 1 ☐ 2 ☐ 3 ☐ 4 ■ 5 ■ 6 ■ 7 ■ 8 ■ 9 ■ 10 ■ 11 ■ 12 ■ 13 ☐ 14 ☐ 15