Q6: AlphaBetaExpinimax

Q6: AlphaBetaExpinimax

Problem Statement

In this question, player A is a minimizer, player B is a maximizer, and C represents a chance node. All children of a chance node are equally likely. Consider a game tree with players A, B, and C. In lecture, we considered how to prune a minimax game tree - in this question, you will consider how to prune an expinimax game tree (like a minimax game tree but with chance nodes). Assume that the children of a node are visited in left-to-right order.

Important: The α-β pruning algorithm does not deal with chance nodes. Instead, for a node n, consider all the values seen so far, and determine whether you can know **without looking at the node** that the value of the node will not affect the value at the top of the tree. If that is the case, then n can be pruned.

For each of the following game trees, give an assignment of terminal values to the leaf nodes (labeled a-h) such that the bolded node can be pruned (it doesn't matter if you prune more nodes). You may give an assignment where an ancestor of the bolded node is pruned (since then the bolded node will never be visited). You **should not** prune on equality, and your terminal values **must** be finite (including negative values).

Important Directions:

- If there exists an assignment where the bolded node can be pruned, enter the utility values of the leaves in the corresponding table.
- If there does **not** exist an assignment where the bolded node can be pruned, enter 'x' (no quotes) in **all** of blanks for the values in the table.

Practice

0 points possible (ungraded)
To make sure you understand how the submission process works, try the following examples out. The following two questions are just for practice and will not count for any points towards the final exam.

1. You found an assignment satisfying the pruning condition and it is:

$$a = 0, b = 1, c = 2, d = 3, e = 4, f = 5, g = 6, h = 7$$

How would you enter this assignment in the following table?

a	b	c	d	e	f

2. You cannot find any assignment that satisfies the pruning condition.

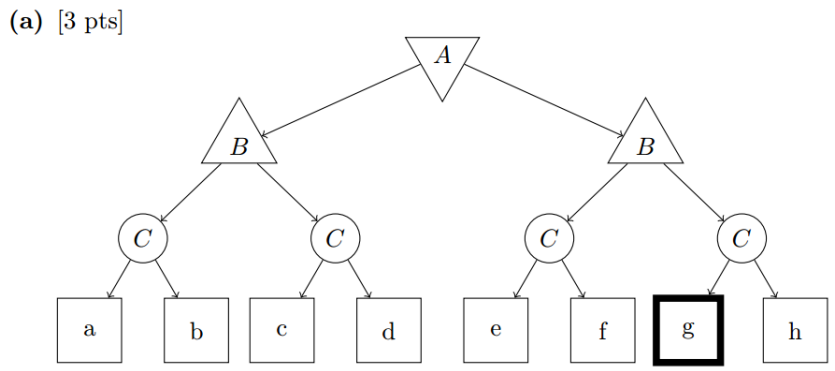
How would you denote that no assignment exists in the following table?

a	b	c	d	e	f

Submit

Part 1

0.0/3.0 points (graded)



a	b	c	d	e	f
1	1	1	1	2	2
Answer: 2	**Answer:** 2	**Answer:** 2	**Answer:** 2	**Answer:** 3	**Answer:** 3

Explanation
In this assignment, the values of the last two leaves do not matter. When we are about to look at the last chance node, we know that the minimizer A can guarantee a score of 2, but the maximizer B can guarantee a score of 3, and so the last chance node is pruned.

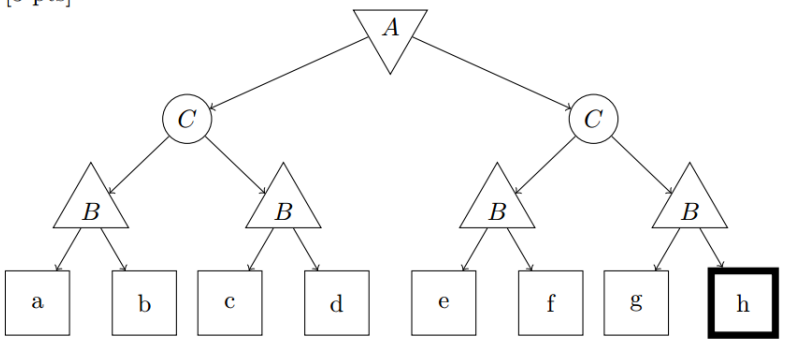
Submit You have used 0 of 2 attempts

Answers are displayed within the problem

Part 2

0.0/3.0 points (graded)

(b) [3 pts]



a	b	c	d	e	f
1	1	1	1	2	2
Answer: 2	**Answer:** 2	**Answer:** 2	**Answer:** 2	**Answer:** 3	**Answer:** 3

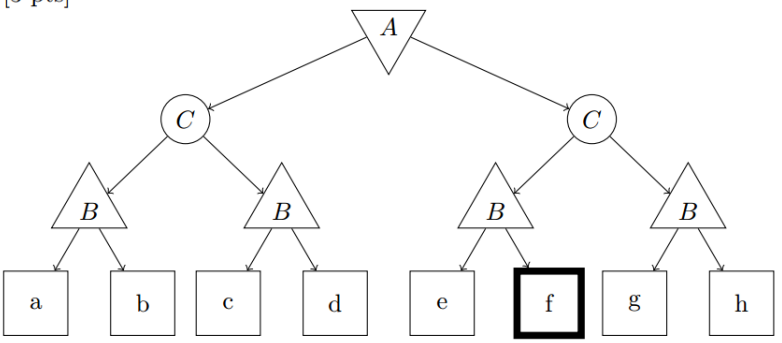
Explanation
In this assignment, the value of the last leaf does not matter. When we are about to look at the bolded node, we know that the minimizer A can guarantee a score of 2, but the second chance node C is guaranteed to have a score of at least 3 (because all of the maximizers below it can guarantee a score of 3).

Submit You have used 0 of 2 attempts

Part 3

0.0/3.0 points (graded)

(c) [3 pts]



a	b	c	d	e	f
x	x	x	x	x	x
Answer: x	**Answer:** x	**Answer:** x	**Answer:** x	**Answer:** x	**Answer:** x

Explanation
Not possible. At the bolded node, the minimizer can guarantee a score of at most the value of its left subtree. However, since we have not visited all the children of C, there is no bound on the value that C could attain. So, we need to continue exploring nodes until we can put a bound on C's value, which means that we have to explore the bolded node.

Submit You have used 0 of 2 attempts