50.021 – Artificial Intelligence

Kwan Hui

Week 9 Theory Homework - Adversarial/Game Search

[The following notes are compiled from various sources such as textbooks, lecture materials, Web resources and are shared for academic purposes only, intended for use by students registered for a specific course. In the interest of brevity, every source is not cited. The compiler of these notes gratefully acknowledges all such sources. ]

Due: 29th March, 11:59pm

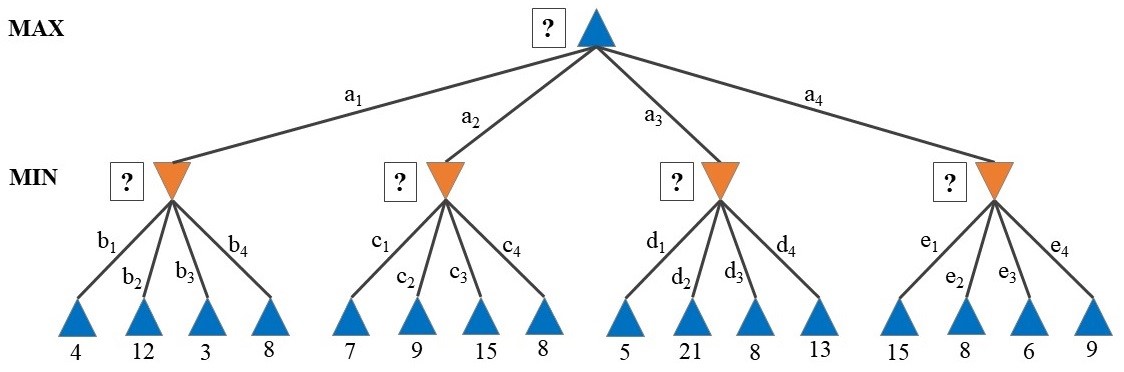
Submission: via eDimension.

**Name: Sharryl Seto**

**Class: Cl03**

**ID: 1005523**

# Minimax Search



Consider the above game tree for a 2-ply game between two players, with the utility scores as listed. Apply the Minimax algorithm on this search, where the exploration of moves/actions is based on alphabetical order (i.e., *a*1*,a*2*,a*3*,...*). Answer the following:

1. List down the Minimax values at each level (i.e., the square boxes with the question mark).

From left to right, the values of the (orange triangles) are: 3, 7, 5, 6.

At the top, the initial state (blue triangle) has the value: 7

1. Briefly explain why those values are chosen.

For this game, being deterministic and fully observable, we want to move to the position with the highest minimax value, so as to get the best achievable payoff against the best play. So for each section, we first choose the smallest utility score. Then we compare them and choose the greatest value for the highest minimax value.

1. Which move/action would be chosen?

a1 b3.

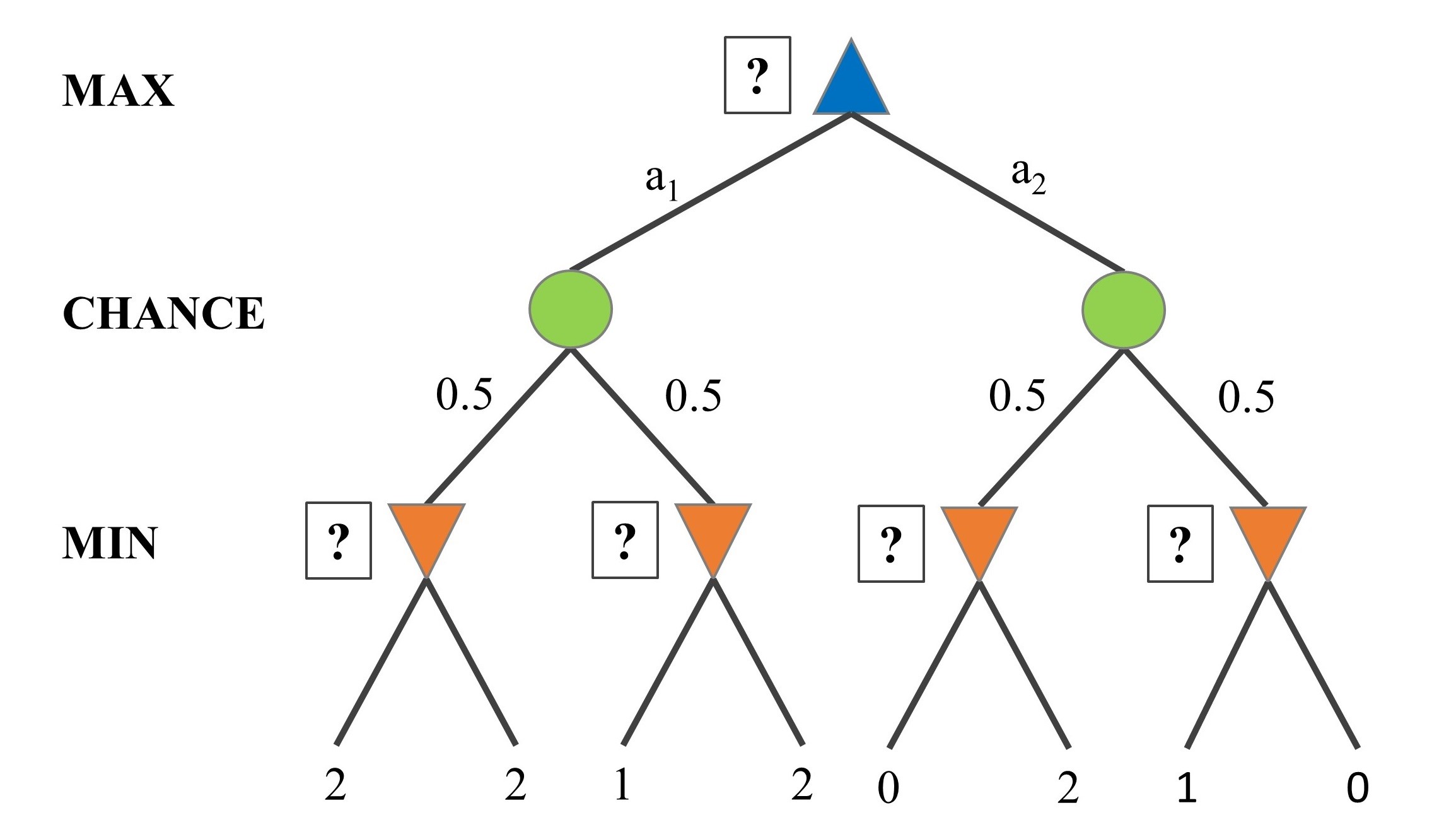
# *α*−*β* Pruning

Using the same search tree from Task 1, apply *α* − *β* pruning with the same move ordering as before. Answer the following:

1. Which moves/actions are pruned (if any)? List them in the order they were removed. None.
2. Briefly explain why these moves/actions were removed (if any).

Nothing was removed because after moving to a2 from a1, *α = 3,*  *β = infinity*. We only prune if *α >= β*, and since the remaining paths are all < 3, nothing needs to be pruned as all paths still need to be checked for the minimum.

# ExpectiMinimax Search



Consider the above game tree for a probabilistic game between two players, with the utility scores as listed at the leaf nodes. In this game, there are chance nodes (denoted by circles) that are based on a fair coin toss. Apply the ExpectiMinimax algorithm on this search, where the exploration of moves/actions is based on alphabetical order. Answer the following:

1. List down the ExpectiMinimax values at each level (i.e., the square boxes with the question mark).

Bottom up:

* (orange triangles) Left to right: 2, 1, 0, 0
* (green circle) Left to right: 1.5, 0
* (blue triangle) top: 1.5

1. Which move/action would be chosen?

a1.