



Bookmarks

- ▶ Artificial Intelligence Course: Getting Started
- ▶ Week 1: Introduction to AI
- ▶ Week 2: Intelligent Agents and Uninformed Search
- ▶ Week 3: Heuristic Search
- ▼ **Week 4: Adversarial Search and Games**

## Week 4 Suggested Readings

4.1 Adversarial Search and Games

4.2 Minimax algorithm

4.3 Alpha-Beta Pruning

4.4 Stochastic games

**Week 4 Quiz: Adversarial Search and Games**

Quiz due Apr 10, 2017 05:30 IST

Week 4: Adversarial Search and Games &gt; Week 4 Quiz: Adversarial Search and Games &gt; Week 4 Quiz: Adversarial Search and Games

## Week 4 Quiz: Adversarial Search and Games

Bookmark this page

### Zero-sum games

10/10 points (graded)

Check all that apply. In zero-sum games:

☒ One agent maximizes one single value, while the other minimizes it. ☐ Each agent tries to maximize independent values.☒ Agents take turns. ☐ Each agent helps the other one win the game.

Submit

You have used 1 of 2 attempts


Correct (10/10 points)

### Minimax

10/10 points (graded)

The minimax algorithm uses a recursive computation of the minimax values of each successor state. The recursion proceeds all the way down to the leaves of the tree. The minimax values are then backed up through the tree.

☐ False☒ True

**Week 4 Project:  
Adversarial Search  
and Games**Project due Apr 11,  
2017 05:00 IST **Week 4 Discussion  
Questions****► Week 5:  
Machine  
Learning 1****Submit**

You have used 1 of 1 attempt

**✓** Correct (10/10 points)**Adversarial search**

0/10 points (graded)

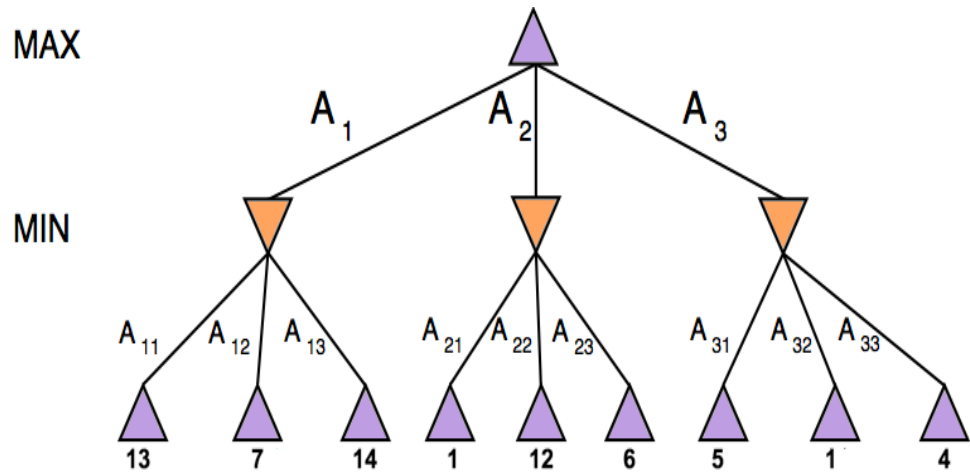
Check all that apply

☒ We can't always search the leaves in game trees because we are limited in time☒ Both Minimax and alpha-beta pruning adopt a depth first search strategy☐ Searching using minimax without evaluation function has to go all the way down to the leaves except when alpha-beta pruning is used.☒ Searching using minimax without evaluation function has to go all the way down to the leaves even when alpha-beta pruning is used.**Submit**

You have used 1 of 1 attempt

**✗** Incorrect (0/10 points)

Consider the following search tree.



### Minimax

10/10 points (graded)

Using minimax, which of the three possible moves should MAX take at the root node?

☒ A1 ✓

☐ A2

☐ A3

Submit

You have used 1 of 1 attempt

✓ Correct (10/10 points)

### Minimax

10/10 points (graded)

Using minimax, what is the value of MAX at the root?

☐ 5


☐ 14

☒ 7 ✓

☐ 12





Submit

You have used 1 of 1 attempt

 Correct (10/10 points)**Minimax with alpha-beta pruning**

10/10 points (graded)

Using minimax with alpha-beta pruning, what branches are pruned? Check all that apply:

☐ A1☐ A2☐ A3☐ A11☐ A12☐ A13☐ A21☒ A22 ☒ A23 ☐ A31☒ A32 ☒ A33 

Submit

You have used 1 of 2 attempts

✓ Correct (10/10 points)

### Alpha-beta pruning

10/10 points (graded)

Check all that apply regarding Alpha-beta pruning:

☒ Both Alpha and Beta are sent down the tree ✓

☐ Min updates Alpha and Max updates Beta

☒ Min updates Beta and Max updates Alpha ✓

☐ Min and Max update both Alpha and Beta

☒ Alpha is the current lower bound on MAX's outcome and Beta is the current upper bound on MIN's outcome ✓

☒ MAX will never choose a move that could lead to a worse score (for MAX) than Alpha. ✓

☐ MIN will always choose a move that could lead to a better score (for MAX) than Beta.

☐ Prune remaining branches at a node whenever Alpha is less than Beta

✓

Submit

You have used 1 of 2 attempts

✓ Correct (10/10 points)

### Move ordering

10/10 points (graded)

When the best moves are on the left of the game tree, the search can go twice as deep as minimax in the same amount of time:

☒ True ✓☐ False

Submit

You have used 1 of 1 attempt

✓ Correct (10/10 points)

### Utility and Eval functions

0/10 points (graded)

The utility function  $utility(s)$  is replaced by an evaluation function  $eval(s)$  to estimate the value of the current state  $s$ . Check all that apply:

☒ Because there is a time constraint for each move☒ It turns non-terminal nodes into terminal leaves☐ The evaluation function does not have to rank terminal states in the same way as the true utility function

Submit

You have used 1 of 1 attempt

✗ Incorrect (0/10 points)

### Expectiminimax

10/10 points (graded)

Comparing Expectiminimax to Minimax, check all that apply:

☒ Expectiminax generalizes Minimax to handle chance nodes ✓☒ For a chance node, Expectiminimax returns the expected value of its successors ✓