MULTI-AGENT SEARCH

Lara J. Martin (she/they)
TA: Aydin Ayanzadeh (he)

9/26/2023

CMSC 671

By the end of class today, you will be able to:

1. Predict the resulting utility from minimax search
2. Identify branches that would be pruned with alpha-beta pruning
3. Reason about the ethics of search algorithms "in the wild"

Modified from slides by Dr. Cassandra Kent & Dr. Cynthia Matuszek

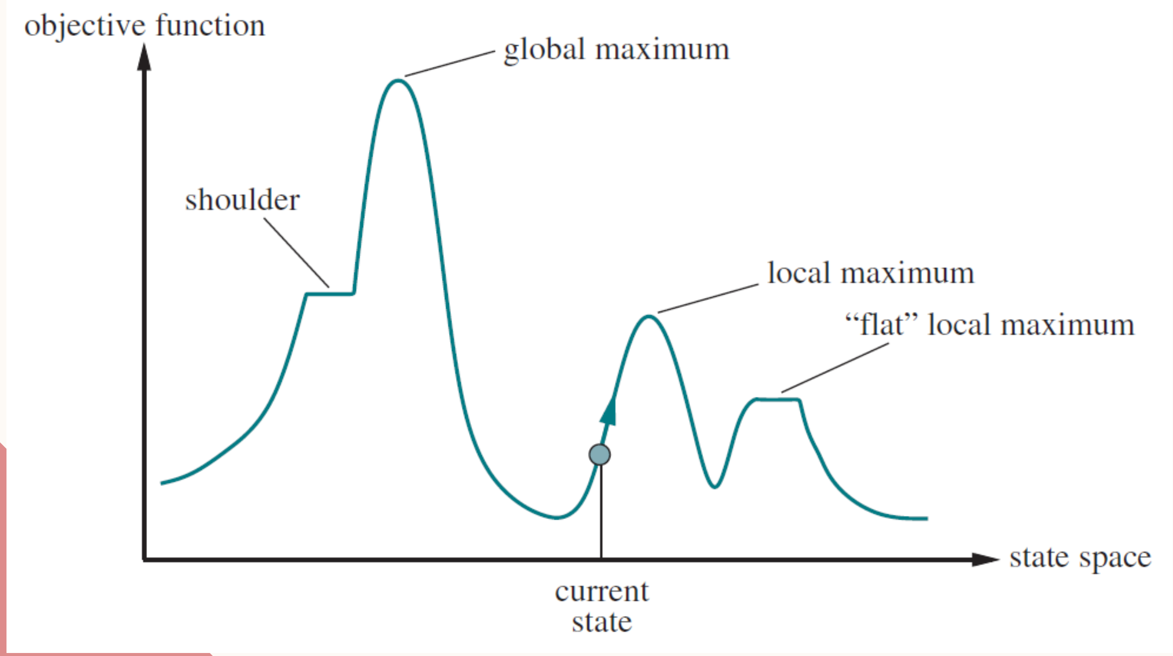
RECAP

LOCAL SEARCH

- What is the goal in local search?
  Find the global maximum
- When do we use local search?
  1) Path doesn't matter
  2) Every state has all the components of the solution

RECAP

LOCAL SEARCH



It's particularly good for continuous state spaces!

ADVERSARIAL SEARCH

ADVERSARIAL SEARCH

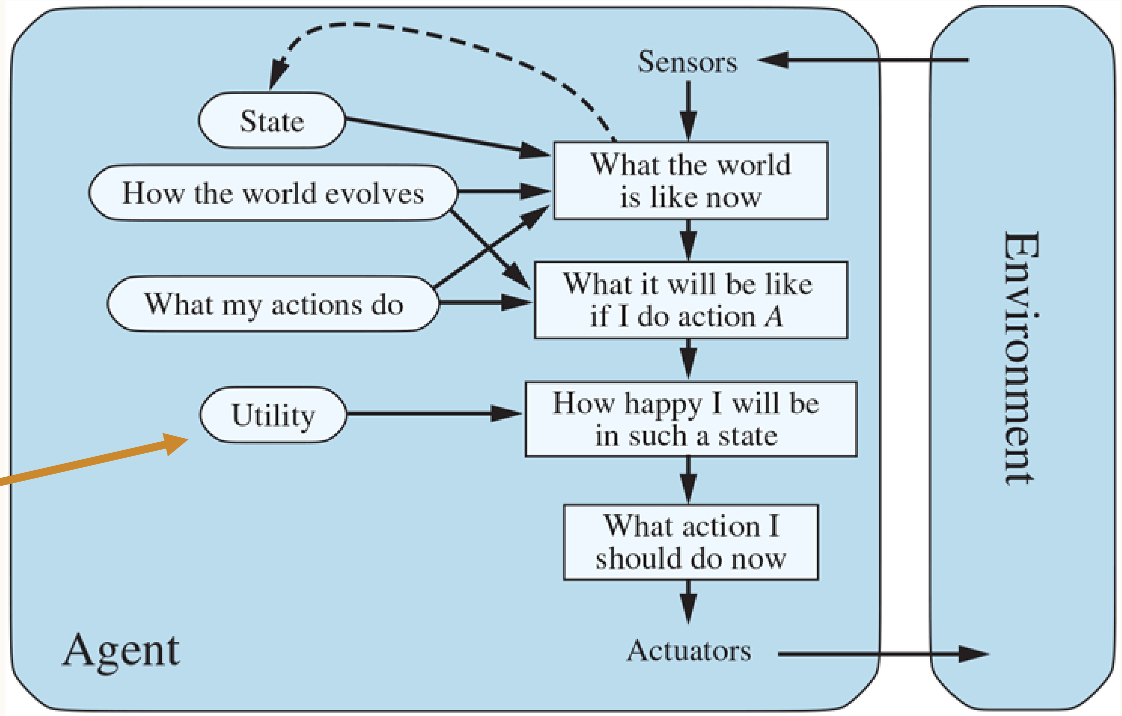
Garry Kasparov vs. Deep Blue, 1997

Lee Sedol vs. AlphaGo, 2016

We can solve multi-agent games with search, with a few assumptions:

- **Two player**, **turn-based** games: multi-agent, static
- **Perfect information**: all players know exact state of the game; fully observable, deterministic
- **Zero-sum**: game result for one player is exactly opposite of the opponent

WHAT IS UTILITY?

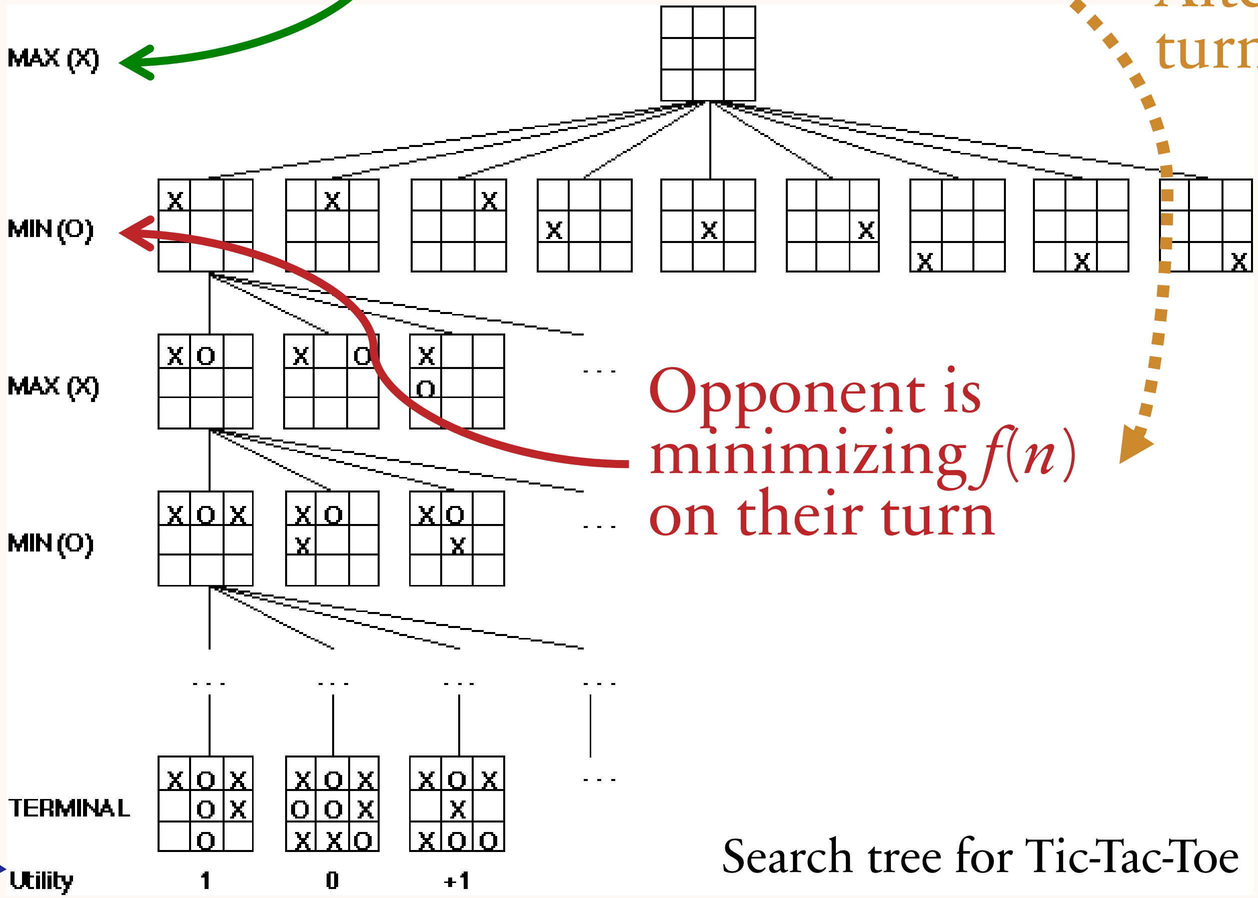


Game score

2-PLAYER SEARCH TREES: MINIMAX

I am maximizing $f(n)$ on my turn

Alternating turns



Opponent is minimizing $f(n)$ on their turn

Did we win, lose, or tie?

Search tree for Tic-Tac-Toe

MINIMAX: PSEUDOCODE

```
function MINIMAX-SEARCH(game, state) returns an action
  player ← game.TO-MOVE(state)
  value, move ← MAX-VALUE(game, state)
  return move

function MAX-VALUE(game, state) returns a (utility, move) pair
  if game.IS-TERMINAL(state) then return game.UTILITY(state, player), null
  v ← −∞
  for each a in game.ACTIONS(state) do
    v2, a2 ← MIN-VALUE(game, game.RESULT(state, a))
    if v2 > v then
      v, move ← v2, a
  return v, move

function MIN-VALUE(game, state) returns a (utility, move) pair
  if game.IS-TERMINAL(state) then return game.UTILITY(state, player), null
  v ← +∞
  for each a in game.ACTIONS(state) do
    v2, a2 ← MAX-VALUE(game, game.RESULT(state, a))
    if v2 < v then
      v, move ← v2, a
  return v, move
```

Find the max value move on our turn

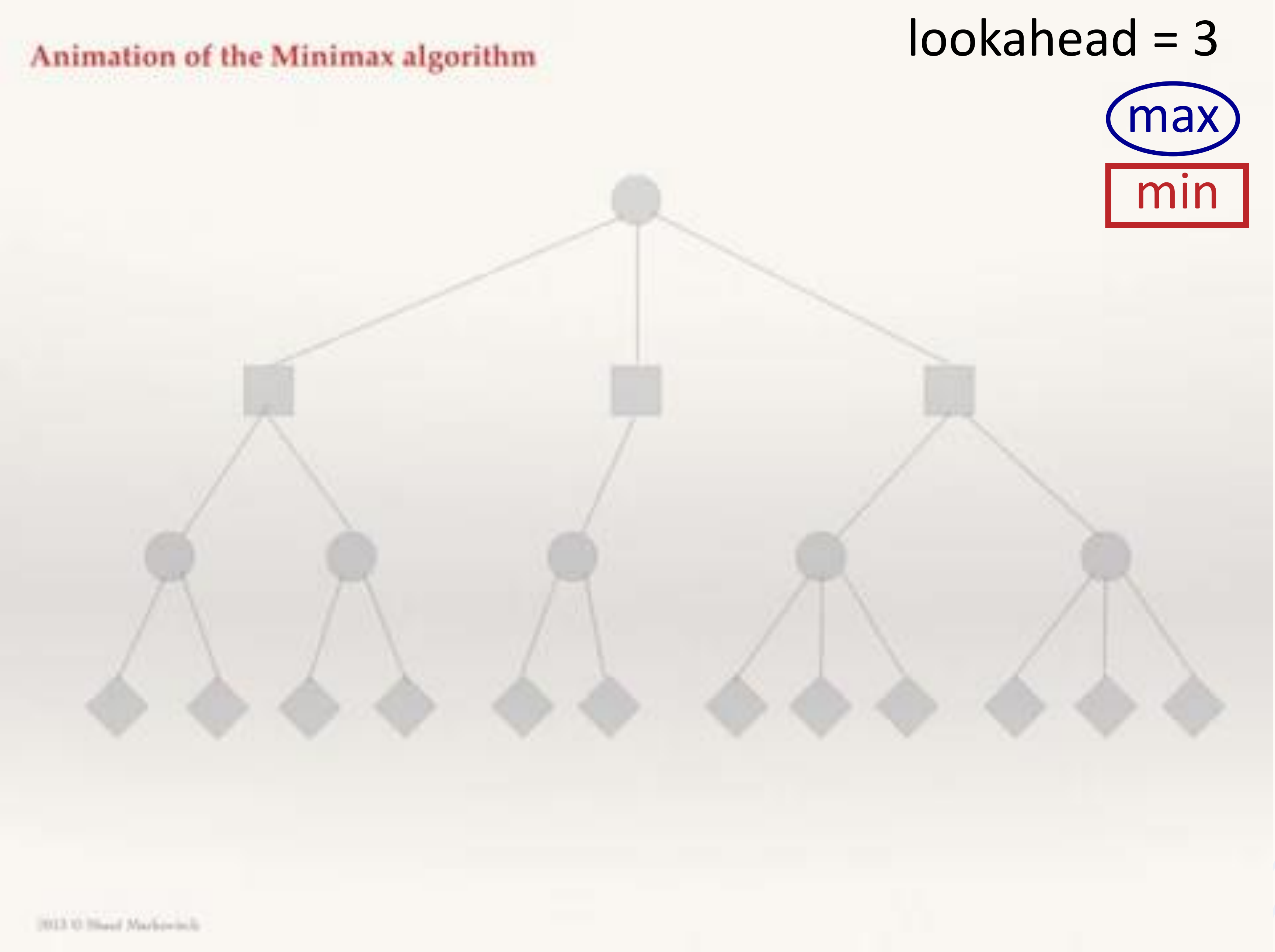
Find the min value move on opponent's future turn

Select max move

Find the max value move on our future turn

Select min move

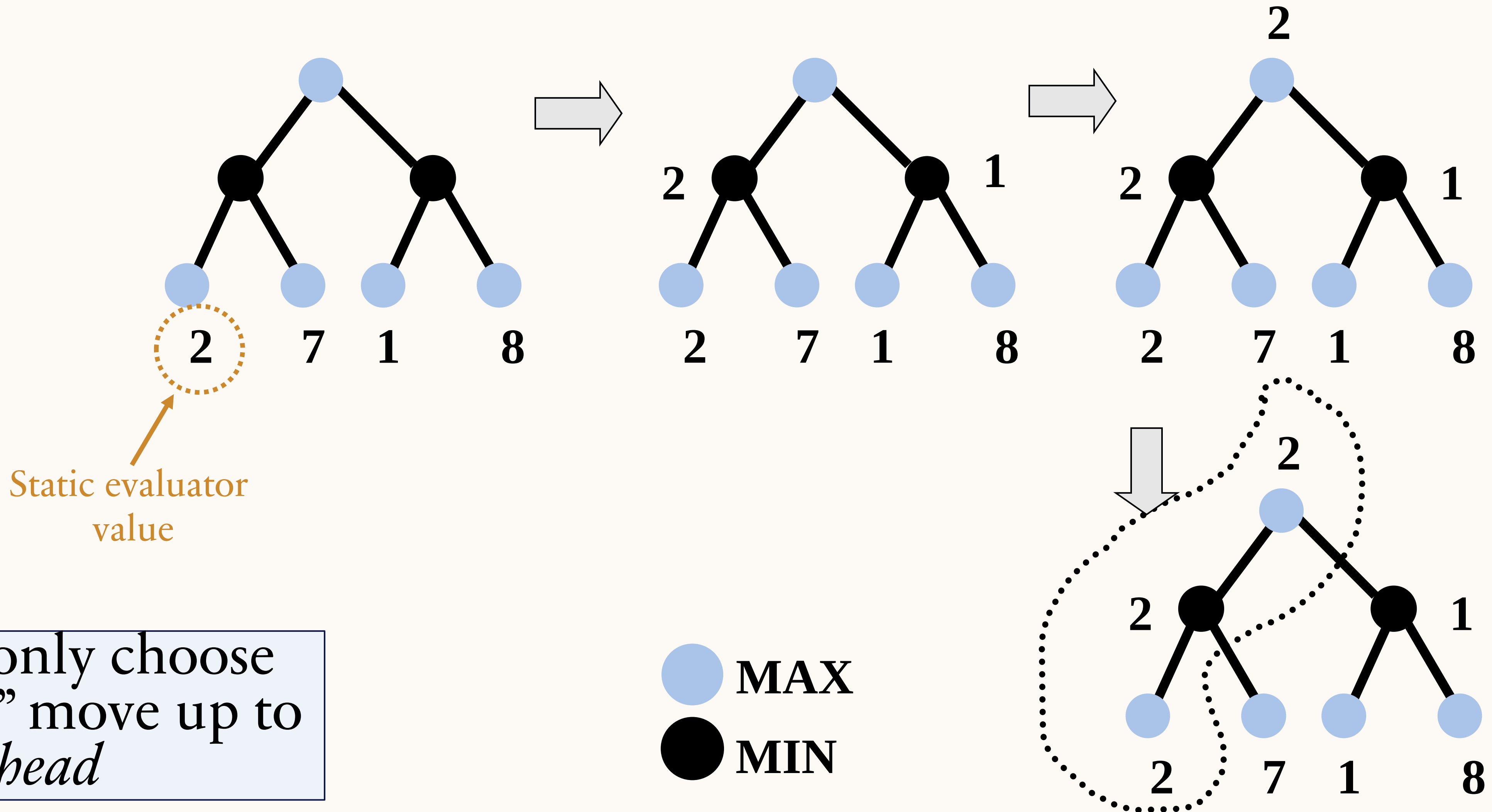
Calculate utility (game result) at terminal nodes lookahead = 3



Calculating bottom-up

https://www.youtube.com/watch?v=6ELUvkSkCts

MINIMAX ALGORITHM



Can only choose "best" move up to *lookahead*

PROPERTIES OF MINIMAX

Very similar to DFS!

Minimax

- Complete (for finite trees)
- Optimal
- $O(b^m)$ time complexity
- $O(bm)$ space complexity

Depth-First Search

- Complete (for finite trees)
- Not optimal
- $O(b^m)$ time complexity
- $O(bm)$ space complexity

Branching factor b: the number of actions the agent can take at any state

Maximum depth of tree m

MINIMAX FOR CHESS

Minimax

- Complete (for finite trees)
- Optimal
- $O(b^m)$ time complexity
- $O(bm)$ space complexity

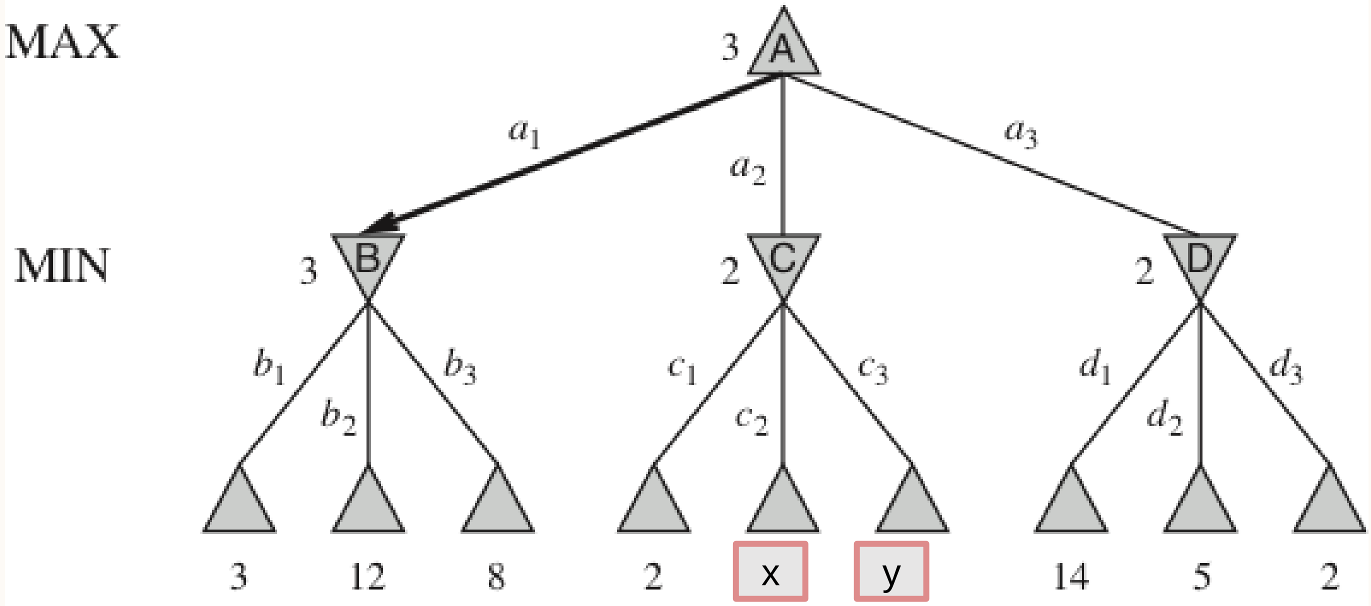
Properties for Chess:

- $b \cong 35$
- $m \cong 100$
- $b^m = 2.55 \times 10^{154}$
- We can not search the full tree!

Branching factor b: the number of actions the agent can take at any state

Maximum depth of tree m

DO WE NEED TO EXPAND EVERY NODE?



$$
\begin{aligned}
\textsc{Minimax}(root) &= \max(\min(3,12,8), \min(2,x,y), \min(14,5,2)) \\
&= \max(3, \min(2,x,y), 2) \\
&= \max(3, z, 2) \qquad \text{where } z = \min(2,x,y) \leq 2 \\
&= 3
\end{aligned}
$$

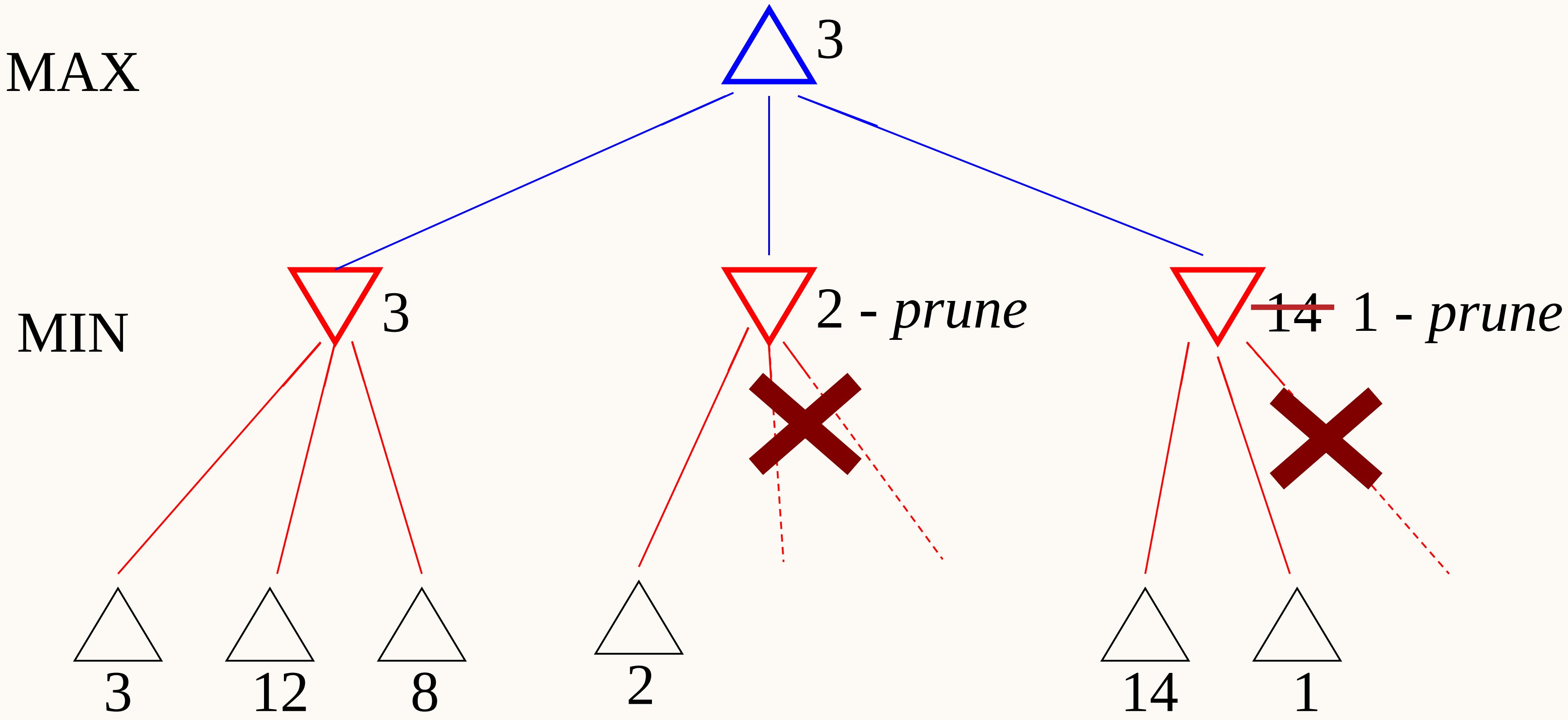
ALPHA-BETA PRUNING

MINIMAX WITH ALPHA-BETA PRUNING

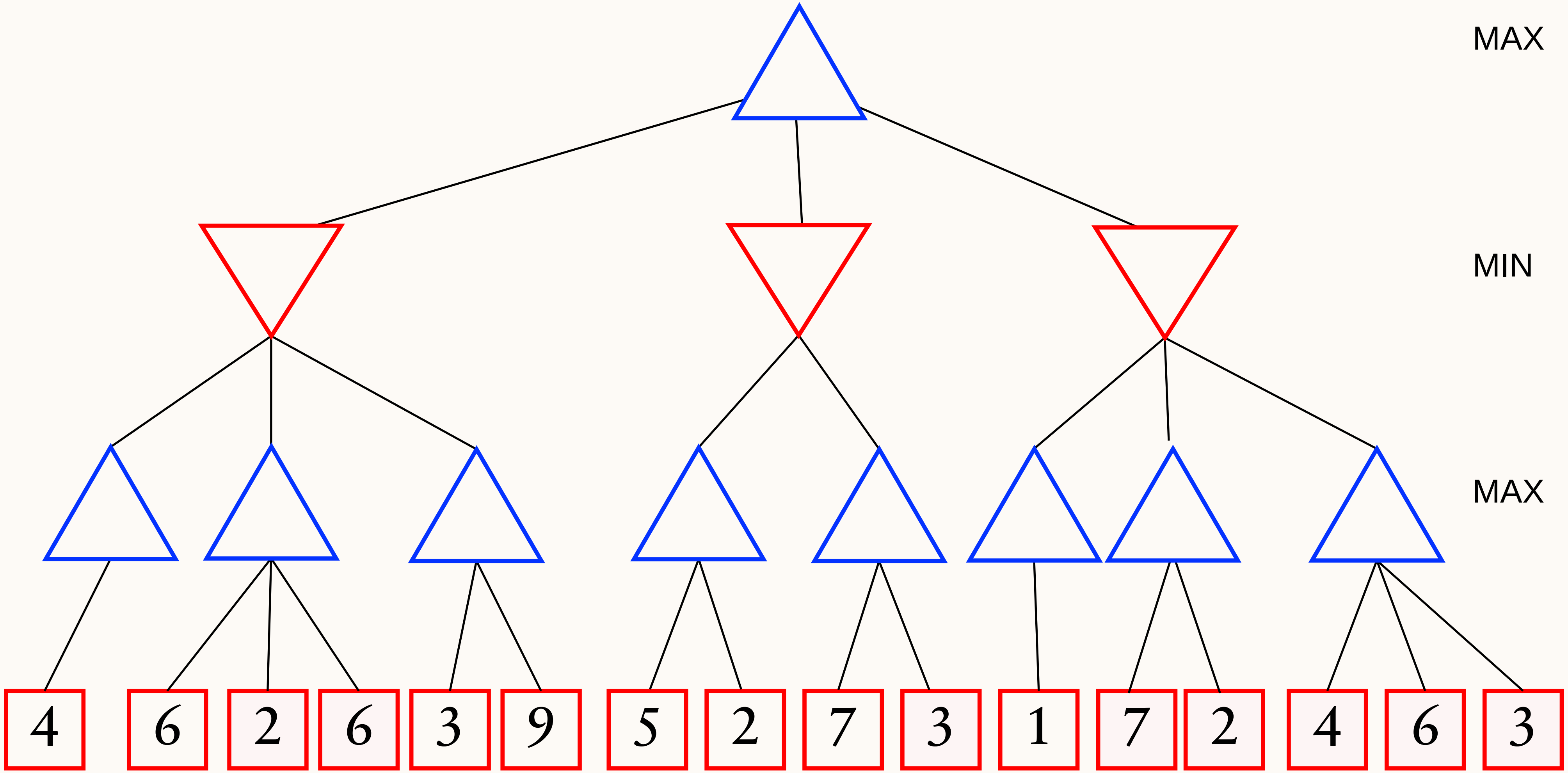
Maintain bounds at each node representing the highest and lowest values, cut off search outside of bounds

- Traverse search tree in *depth-first order*
- At each **MAX** node n, $\alpha(n)$ = maximum value found so far
- At each **MIN** node n, $\beta(n)$ = minimum value found so far
  - α starts at $-\infty$ and increases, β starts at $+\infty$ and decreases
- If $\alpha > \beta$, **prune the branch**
  - i.e., stop searching that node's successors

ALPHA-BETA EXAMPLE (B=3)



YOUR TURN: ALPHA-BETA PRUNING



EFFECTS OF ALPHA-BETA PRUNING

Branching factor b
Depth searched d

- Alpha-beta is guaranteed to:
  - Compute the same value for the root node as minimax (Pruning will not affect optimality)
  - With ≤ computation
- Worst case time complexity: $O(b^d)$ - same as `Minimax`
  - Nothing pruned
- Best case time complexity: $O(b^{d/2})$ – effective branching factor becomes $\sqrt{b}$
  - When each player's best move is the first alternative generated

For chess:

- $b \cong 6$ instead of $b \cong 35$
- Lets us search **twice as deep**!

In Deep Blue, empirically, alpha-beta pruning took average branching factor from ~35 to ~6!

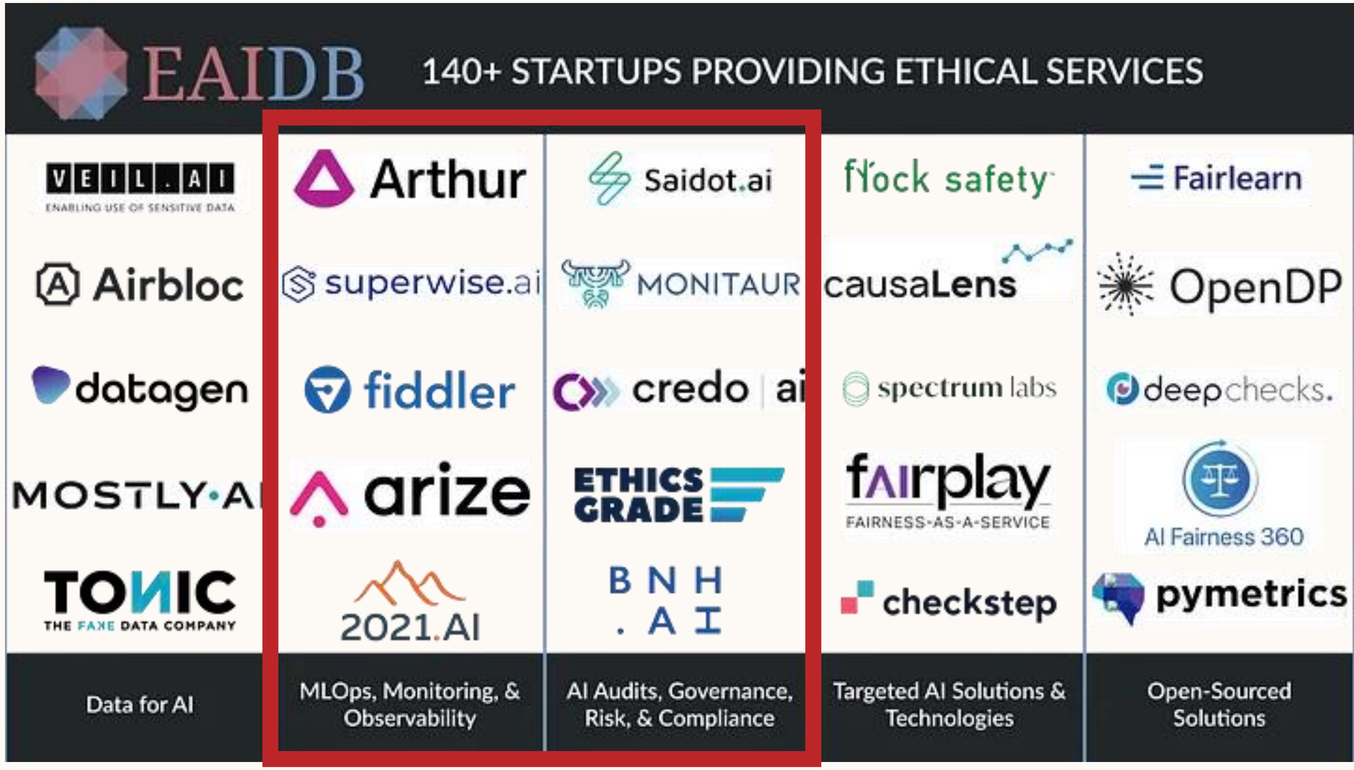
ETHICS OF SEARCH

AI ETHICS: SEARCH

- So far, most of the lectures talked about abstract algorithms
- But now with adversarial search, we can see where humans can be involved
- Should we consider ethical issues, even for abstract, game-playing algorithms such as minimax search?

A FRAMEWORK FOR IDENTIFYING ETHICAL ISSUES IN AI

- Many groups are being formed to develop methods for identifying and discussing ethical issues in AI
- We'll build up a question-based framework to help identify possible ethical issues with the approaches we're discussing in class
- You can then use these questions to help you think about these issues in future projects



https://medium.com/@abhinavr2121/the-ethical-ai-ecosystem-market-map-39779a9ea4ce

EXPLAINABILITY AND INTERPRETABILITY

Explainability: the degree to which we can understand the decisions made by an AI agent

- Improve human trust in AI systems
- Prevent AI "cheating", evaluate value alignment, identify biases
- Further scientific knowledge
- Legal situations to evaluate safety and establish accountability

Interpretability: ability to explain or to present in understandable terms to a human[1]

[1]Towards A Rigorous Science of Interpretable Machine Learning, Doshi-Velez and Kim, 2017.

EXPLAINABILITY AND INTERPRETABILITY

- How clear is our agent’s decision making? Is it transparent or is it a black box?
- Can we make changes to the algorithm to make its decisions more explainable?
- Can we develop tools that make the algorithm’s decisions easier to interpret?

EXPLAINABILITY AND INTERPRETABILITY - MINIMAX

- How clear is our agent's decision making? Is it transparent or is it a black box?
  - It's very transparent, we know the objective function and how it's optimized
- Can we make changes to the algorithm to make its decisions more explainable?
  - Return objective function value with the decision, return alternatives, indicate whether it used approximation or exact evaluation
- Can we develop tools that make the algorithm's decisions easier to interpret?
  - Example: DeepMind's AlphaGo Teaching Tool

INEQUALITY

- Who has access to this AI agent?
  - Could this create new inequality between groups that have access and do not have access?
- Is this system reinforcing existing structures that create inequality?
  - If yes, is there regulation for this technology that can prevent this?

INEQUALITY - MINIMAX

- Who has access to this AI agent?
  - Could this create new inequality between groups that have access and do not have access?
  - Example: DeepMind's AlphaGo Teaching Tool was made available online to everyone for free
- Is this system reinforcing existing structures that create inequality?
  - If yes, is there regulation for this technology that can prevent this?
  - Example: Impact on tournaments with prizes – potential to use AI to cheat, need systems in place to detect and prevent this

JOB DISPLACEMENT

Automation replacing human workers is a major concern for robotics.

Also a growing issue in AI, as our algorithms and models start outperforming human experts.

- Will this algorithm displace human workers?
  - If yes, is there a plan in place to help those displaced workers?

- Will this algorithm/agent create new jobs? Who will benefit?

MORE EXAMPLES AND DISCUSSION FROM GO

Example of how AI has changed professional Go:

My opponent has remembered the variations in the tool thoroughly by heart*, and this gives me headache…. **After the tool is published, the difference in strength of professional players has been shortened**….(In the fifth match) I decided that I need to **play differently than AlphaGo suggested, so I played moves from another AI**, combined with my own thoughts. However I feel that my opening in the fifth match was not successful, and **AIs probably would not rate it high**.* - Ke Jie after winning his fifth world title in 2018

MORE EXAMPLES AND DISCUSSION FROM GO

Lee Sedol, the only person to win a game against AlphaGo or AlphaZero, retired in 2019 stating “With the **debut of AI in Go games**, I've realized that I'm not at the top even if I become the number one through frantic efforts…**Even if I become the number one, there is an entity that cannot be defeated**”

On the other hand, AI’s success in Go has invigorated the amateur scene, causing many new players to learn the game.

MORE EXAMPLES AND DISCUSSION FROM CHESS

Positive impact on Chess:

- Chess AI agents, such as Stockfish, are often integrated in resources for learning the game
- AlphaZero recently beat the best version of Stockfish in 2017, invigorated the chess community by revitalizing old strategies and showing you can succeed with “riskier” play
- Garry Kasparov frequently gives talks at AI conferences

FOR NEXT CLASS

- Start reading Chapter 7.1-7.5