CS540 Introduction to Artificial Intelligence
Lecture 19

Young Wu
Based on lecture slides by Jerry Zhu and Yingyu Liang

July 29, 2019

Zero-Sum Games

Motivation

- If the sum of the reward or cost over all players at each terminal state is 0, the game is called a zero-sum game.
- Usually, for games with one winner: the reward for winning and the cost of losing are both 1. If the game ends with a tie, both players get 0.

Tic Tac Toe Example

Motivation

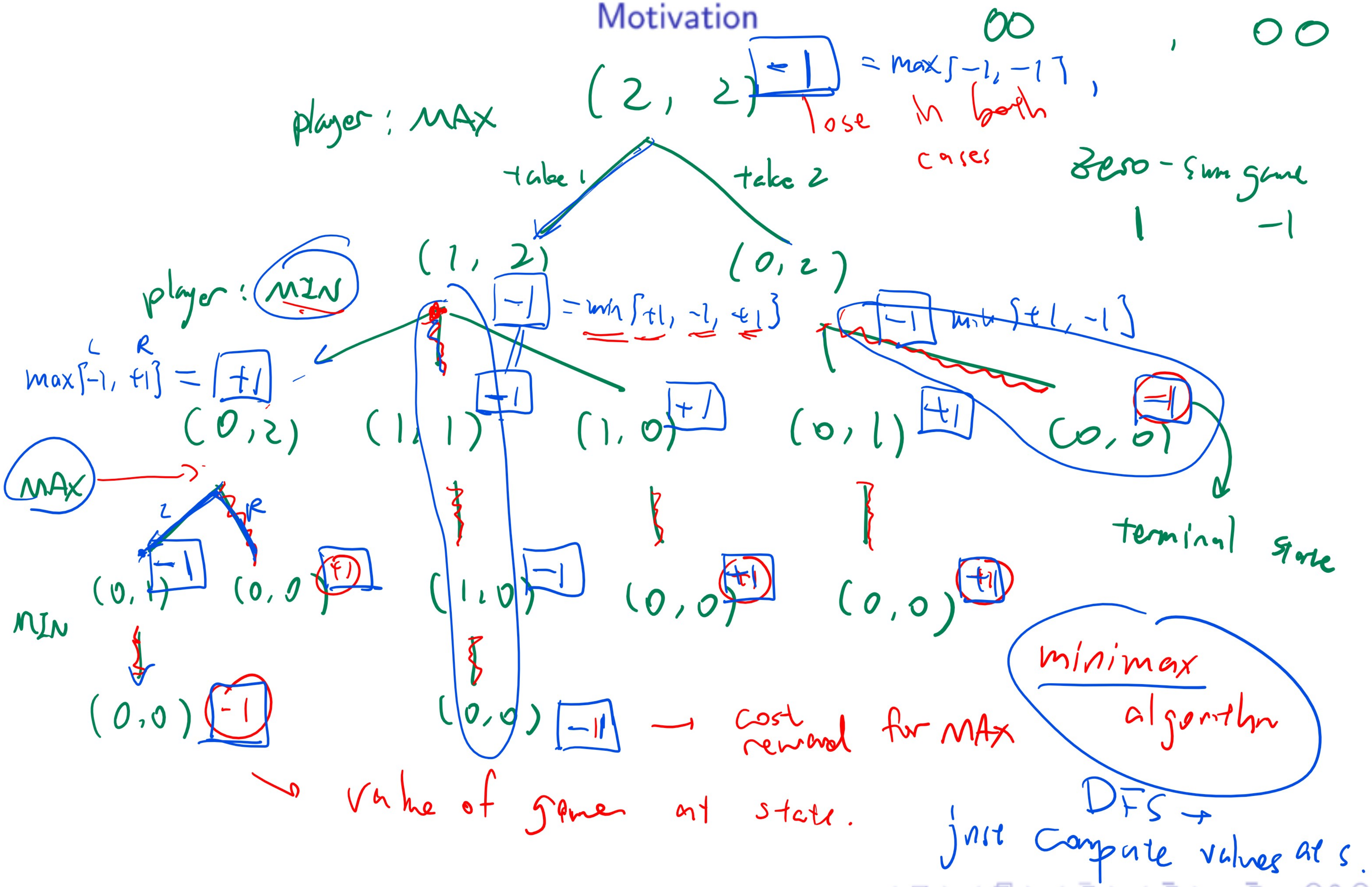
Nim Game Example

Quiz (Graded)

- Ten objects. Pick 1 or 2 each time. Pick the last one to win.
- A: Pick 1.
- B: Pick 2.
- C, D, E: Don't choose.

2 Nim Game Example

Motivation



Minimax Algorithm

Description

- Use DFS on the game tree.

Minimax Algorithm

Algorithm

- Input: a game tree (V, E, c), and the current state s.
- Output: the value of the game at s.
- If s is a terminal state, return $c(s)$.
- If the player is MAX, return the maximum value over all sucessors.

$$\alpha(s) = \max_{s' \in s'(s)} \beta(s')$$

- If the player is MIN, return the minimum value over all sucessors.

$$\beta(s) = \min_{s' \in s'(s)} \alpha(s')$$

Backtracking

Discussion

- The optimal actions (solution paths) can be found by backtracking from all terminal states as in DFS.

$$s^{\star}(s) = \arg \max_{s' \in s'(s)} \beta(s') \text{ for MAX}$$

$$s^{\star}(s) = \arg \min_{s' \in s'(s)} \alpha(s') \text{ for MIN}$$

2 Nim Game Example

Discussion

Minimax Performance

Discussion

- The time and space complexity is the same as DFS. Note that $D = d$ is the maximum depth of the terminal states.

$$T = b + b^2 + ... + b^d$$

$$S = (b - 1) \cdot d$$

Non-deterministic Game

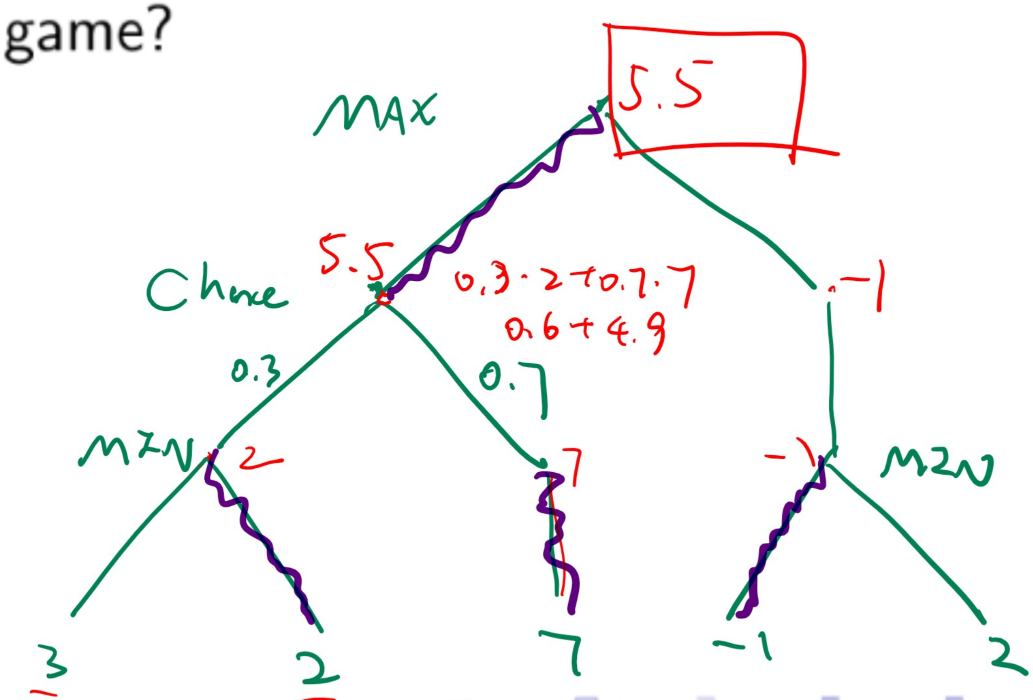
Discussion

- For non-deterministic games in which chance can make a move (dice roll or coin flip), use expected reward or cost instead.
- The algorithm is also called expectiminimax.

Game Tree with Chance Example

Quiz (Graded)

- Fall 2005 Midterm Q7
- Max can pick L or R. If Max picks L, Chance picks L with probability 0.3 and R with probability 0.7. If Chance picks L, Min picks L to get 3, R to get 2, and if Chance picks R, Min gets 7. If Max picks R, Min picks L to get -1 and R to get 2. What is the value of the game?
- A: -1
- B: 2
- C: 5.5
- D: 5.8
- E: 7



Pruning

Motivation

- Time complexity is a problem because the computer usually has a limited amount of time to "think" and make a move.
- It is possible to reduce the time complexity by removing the branches that will not lead the current player to win. It is called the Alpha-Beta pruning.

Alpha Beta Pruning

Description

- During DFS, keep track of both α and β for each vertex.
- Prune the subtree with $\alpha \geqslant \beta$.

Alpha Beta Simple Example

Quiz (Grade)

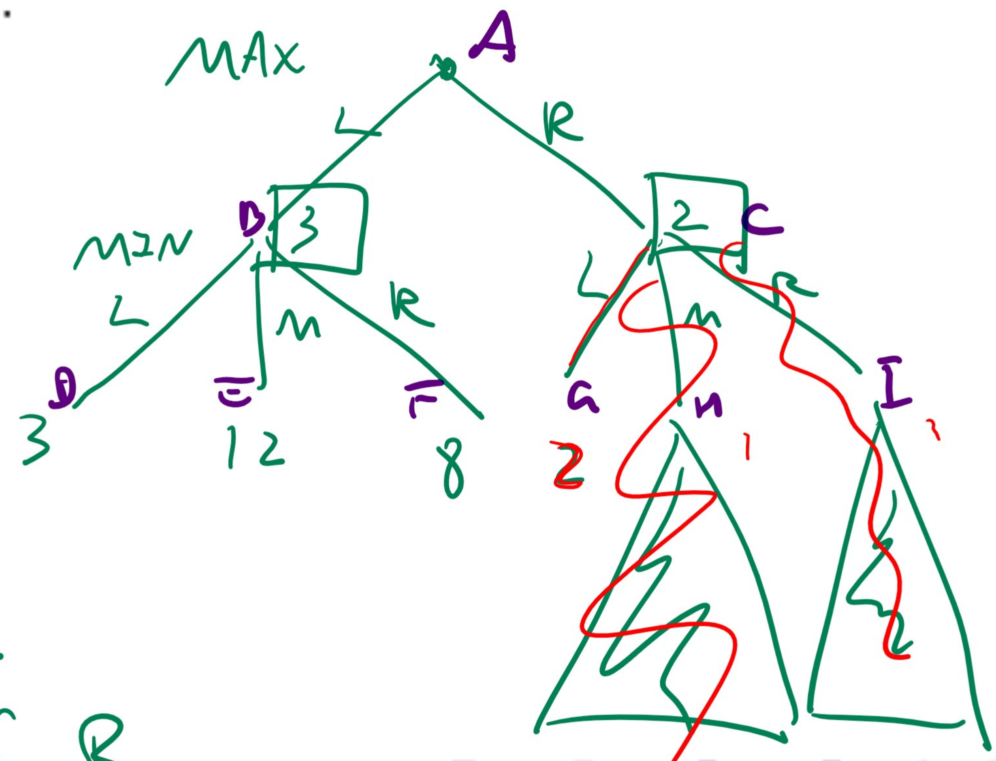
- Fall 2014 Final Q13
- After MAX picks L, MIN can pick L, M, R to get 3, 12, 8. After MAX picks R, MIN can pick L, M, R to get 2, 15, 6. Which vertices can be pruned.
- A: M after L
- B: R after L
- C: L after R
- D: M after R
- E: R after R

G H I D E F B C A ← stack

3 = max {3, ≤ 2} = 3

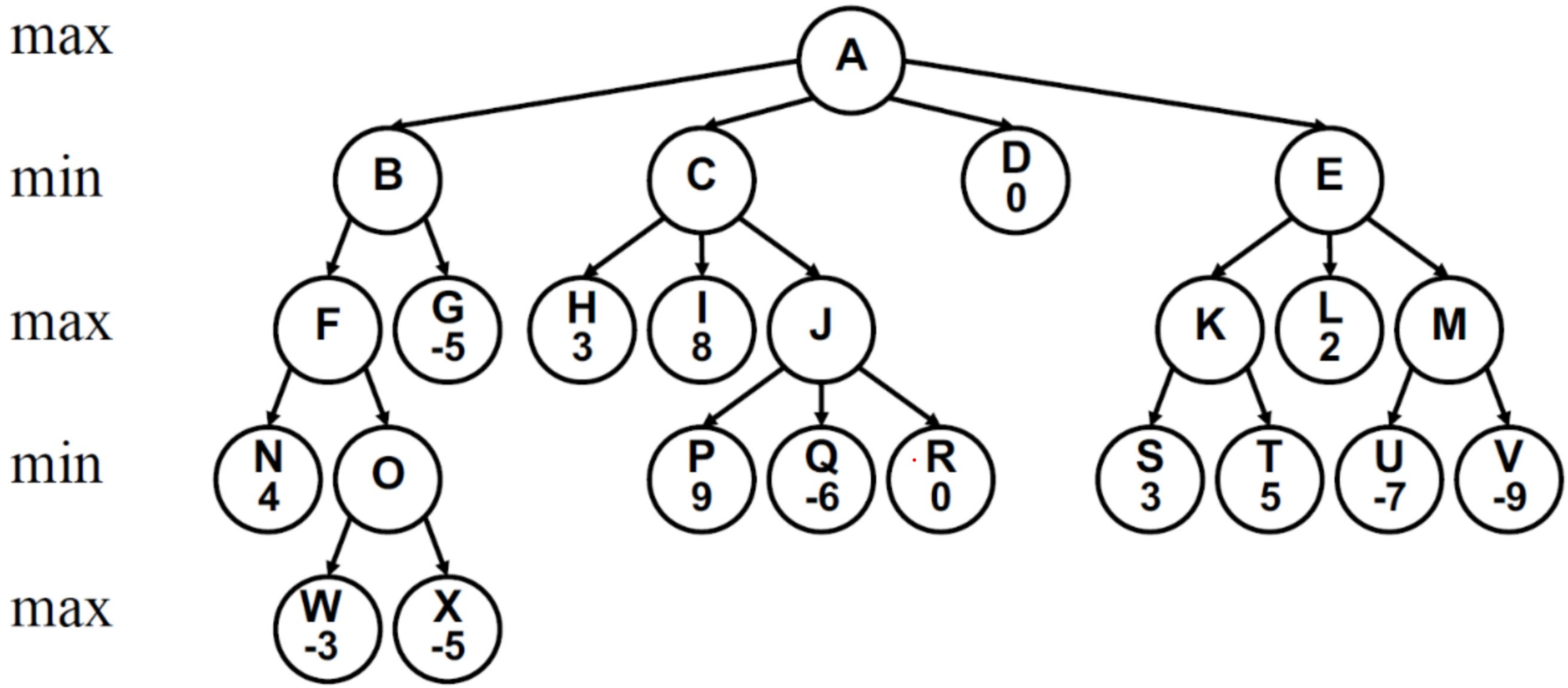
if MAX knows R is bad → does not care how bad

MAX can get 3 if L
can get at most 2 if R



Alpha Beta Example, Part I

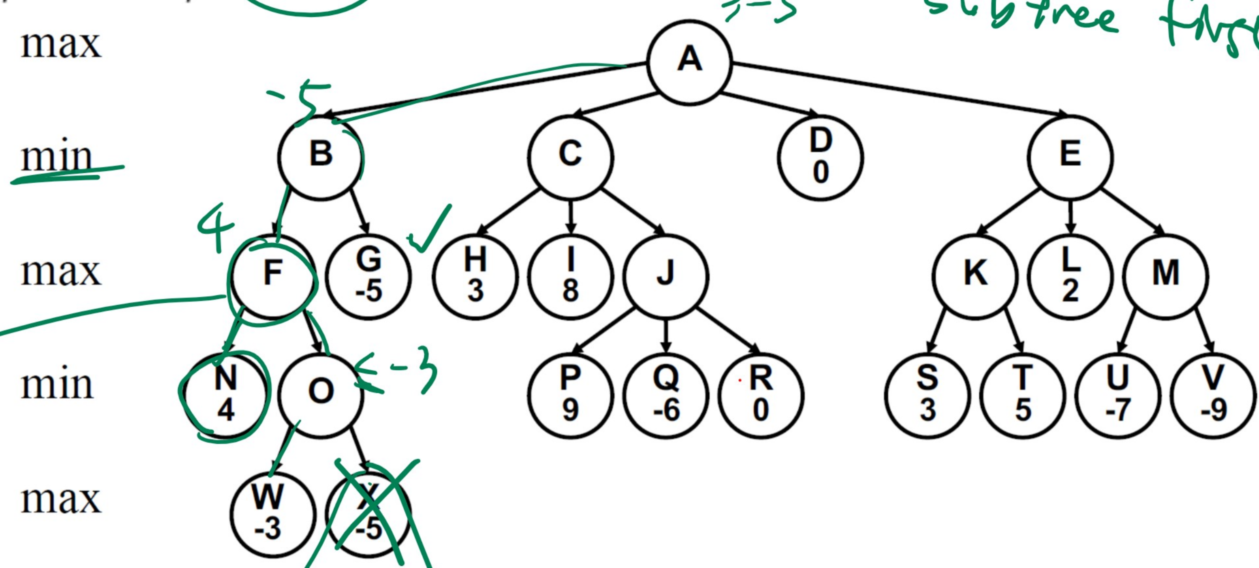
Quiz (Graded)



Alpha Beta Example, Part II

Quiz (Graded)

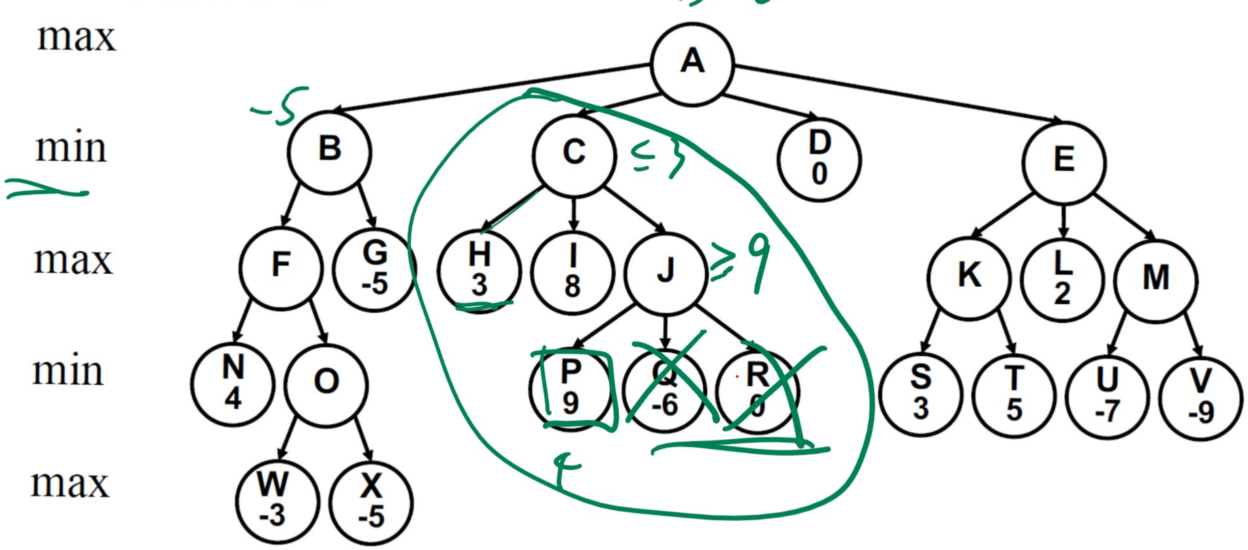
- Which one of the following vertices can be Alpha Beta pruned?
- A: N, B: G, C: O, D: W, E: X



Alpha Beta Example, Part III

Quiz (Graded)

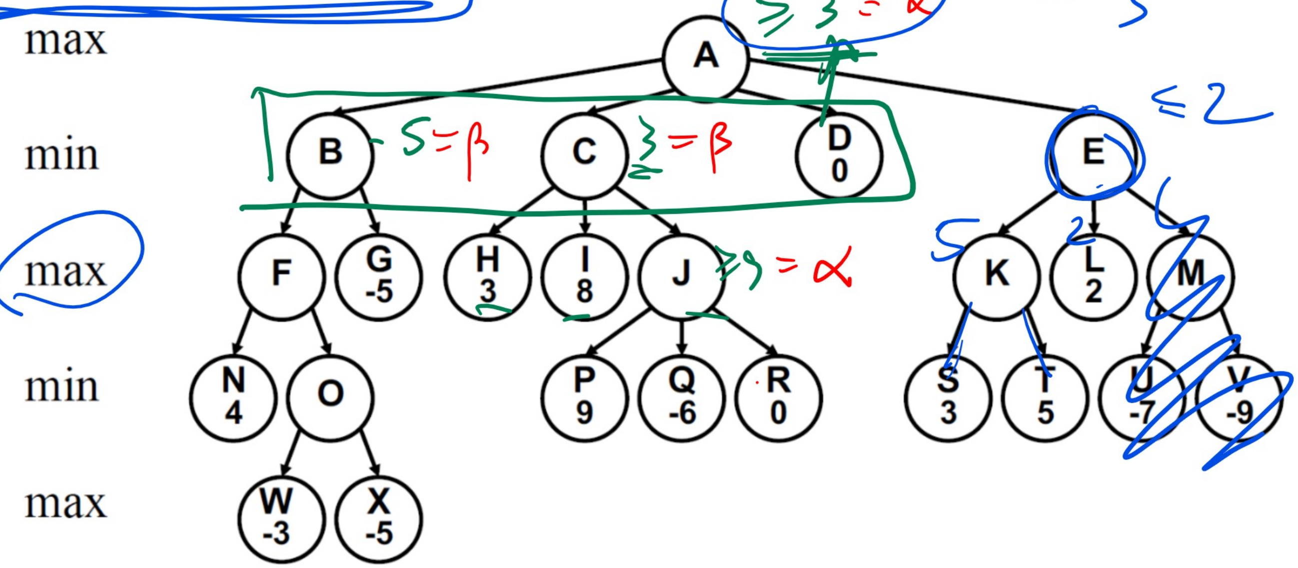
- Which one of the following vertices can be Alpha Beta pruned?
- A: I, B: J, C: P, D: Q, E: R



Alpha Beta Example, Part IV

Quiz (Graded)

- Which one of the following vertices can be Alpha Beta pruned?
- A: T, B: L, C: M, D: U, E: V



Alpha Beta Pruning Algorithm, Part I

Algorithm

- Input: a game tree (V, E, c), and the current state s.
- Output: the value of the game at s.
- If s is a terminal state, return $c(s)$.

Alpha Beta Pruning Algorithm, Part II

Algorithm

- If the player is MAX, return the maximum value over all sucessors.

$$\alpha(s) = \max_{s' \in s'(s)} \beta(s')$$

$$\beta(s) = \beta(\text{ parent } (s))$$

- Stop and return β if $\alpha \geqslant \beta$.

α so far when finding max

- If the player is MIN, return the minimum value over all sucessors.

$$\beta(s) = \min_{s' \in s'(s)} \alpha(s')$$

$$\alpha(s) = \alpha(\text{ parent } (s))$$

- Stop and return α if $\alpha \geqslant \beta$.

β so far when finding min

Alpha Beta Performance

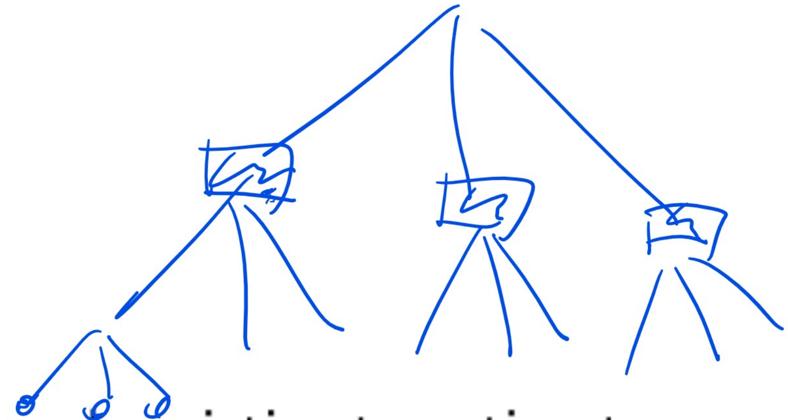
Discussion

- In the best case, the best action of each player is the leftmost child.
- In the worst case, Alpha Beta is the same as minimax.

time
space complexity same.

Static Evaluation Function

Definition



- A static board evaluation function is a heuristics to estimate the value of non-terminal states.
- It should reflect the player's chances of winning from that vertex.
- It should be easy to compute from the board configuration.

Evaluation Function Properties

Definition

- If the SBE for one player is x, then the SBE for the other player should be $-x$.
- The SBE should agree with the cost or reward at terminal vertices.

Linear Evaluation Function Example

Definition

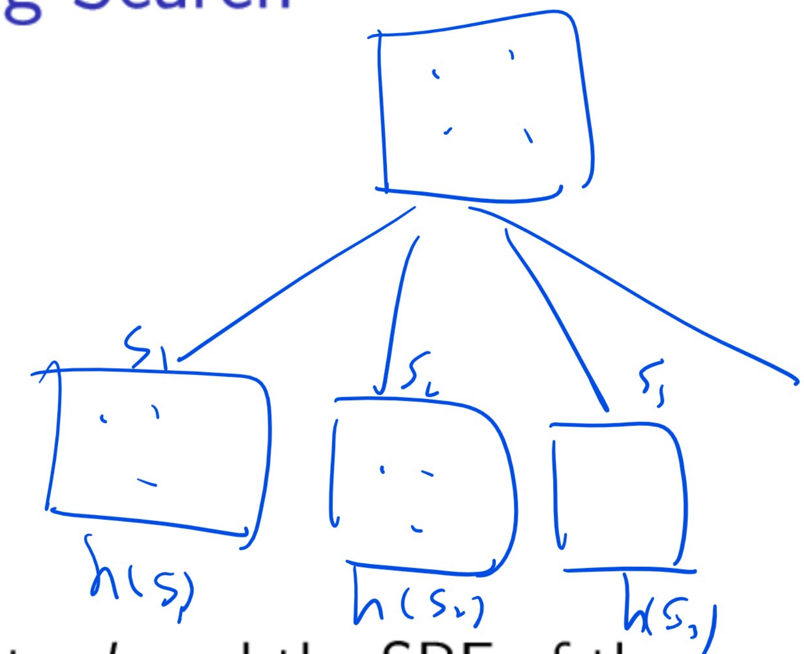
- For Chess, an example of an evaluation function can be a linear combination of the following variables.

1. Material.
2. Mobility.
3. King safety.
4. Center control.

- These are called the features of the board.

Iterative Deepening Search

Discussion

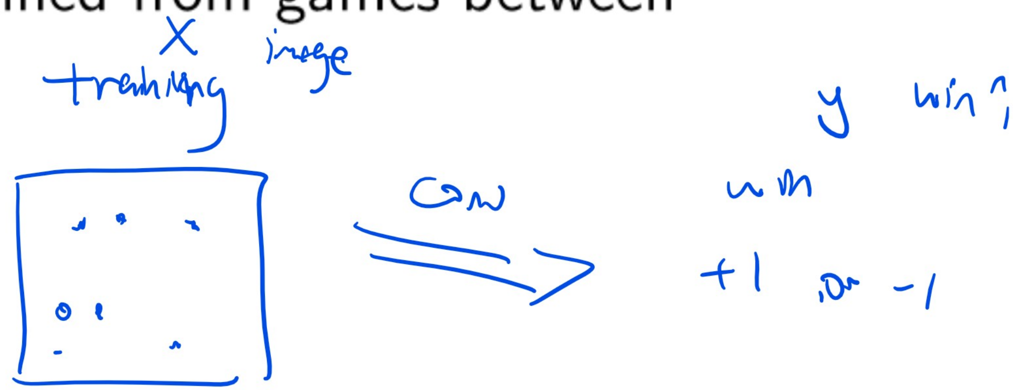


- IDS could be used with SBE.
- In iteration *d*, the depth is limited to *d*, and the SBE of the non-terminal vertices are used as their cost or reward.

Non Linear Evaluation Function

Discussion

- The SBE can be estimated given the features using a neural network.
- The features are constructed using domain knowledge, or a possibly a convolutional neural network.
- The training data are obtained from games between professional players.



Monte Carlo Tree Search

Discussion

- Simulate random games by selecting random moves for both players.
- Exploitation by keeping track of average win rate for each successor from previous searches and picking the successors that lead to more wins.
- Exploration by allowing random choices of unvisited successors.

Monte Carlo Tree Search Diagram

Discussion

Upper Confidence Bound

Discussion

- Combine exploitation and exploration by picking sucessors using upper confidence bound for tree.

$$\frac{w_s}{n_s} + c\sqrt{\frac{\log t}{n_s}}$$

- w_s is the number of wins after successor s, and n_s the number of simulations after successor s, and t is the total number of simulations.
- Similar to the UCB algorithm for MAB.

Alpha GO Example

Discussion

- MCTS with $> 10^5$ playouts.
- Deep neural network to compute SBE.