Identify and connect

Playing and Solving Games

Zero-sum games with perfect information

R&N 6

Types of Games (informal)

	Deterministic	Chance
Perfect Information	Chess, Checkers Go	Backgammon, Monopoly
Imperfect Information	Battleship	Bridge, Poker, Scrabble, wargames

Types of Games (informal)

	Deterministic	Chance
Perfect Information	Chess, Checkers Go	Backgammon, Monopoly
Imperfect Information	Battleship	Bridge, Poker, Scrabble, wargames

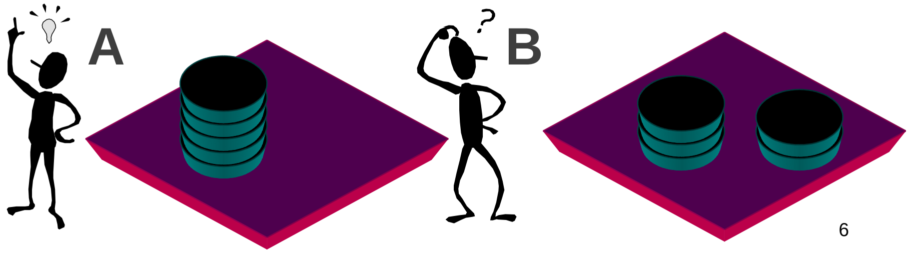
Note: This initial material uses the common definition of what a "game" is. More interesting is the generalization of the theory to scenarios that are far more useful to a wide range of decision making problems.

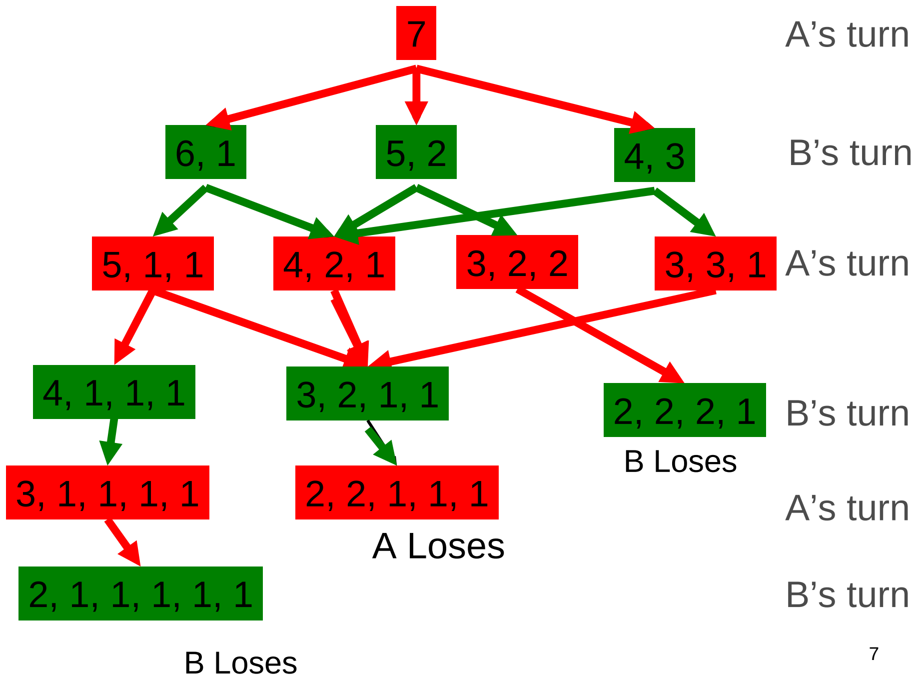
Definitions

- *Two-player game*: Player A and B. Player A starts.
- *Deterministic*: None of the moves/states are subject to chance (no random draws).
- *Perfect information*: Both players see all the states and decisions. Each decision is made *sequentially*.
- *Zero-sum*: Player's A gain is exactly equal to player B's loss. One of the player's must win or there is a draw (both gains are equal).

Example

- Initially a stack of pennies stands between two players
- Each player divides one of the current stacks into two unequal stacks.
- The game ends when every stack contains one or two pennies
- The first player who cannot play loses





7 A's turn

6, 1 5, 2 4, 3 B's turn

5, 1, 1 4, 2, 1 3, 2, 2 3, 3, 1 A's turn

4, 1, 1, 1 3, 2, 1, 1 2, 2, 2, 1 B's turn

B Loses

3, 1, 1, 1, 1 2, 2, 1, 1, 1 A's turn

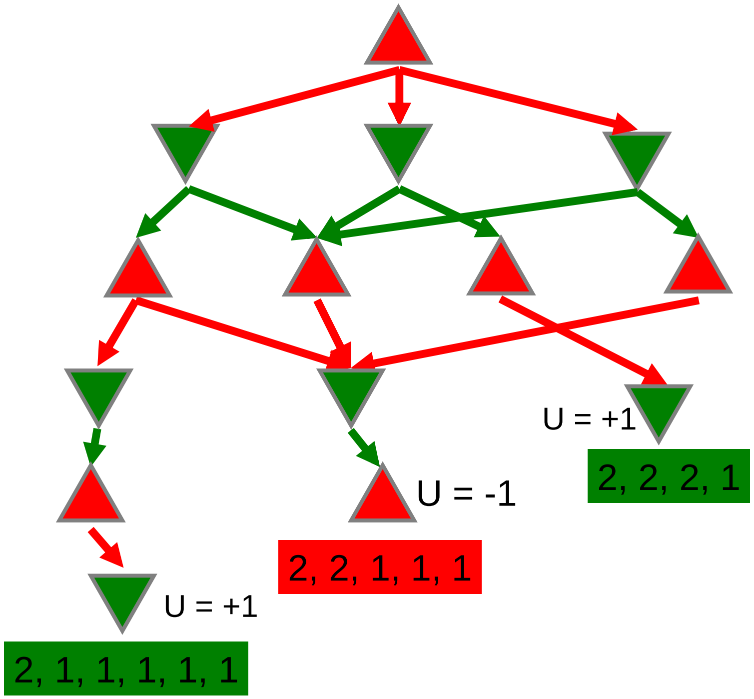
A Loses

2, 1, 1, 1, 1, 1 B's turn

B Loses

Search Problem

- *States*: Board configuration + next player to move
- *Successor*: List of states that can be reached from the current state through of legal moves
- *Terminal state*: States at which the games ends
- *Payoff/Utility*: Numerical value assigned to each terminal state. Example:
  - U(s) = +1 for A win, -1 for B win, 0 for draw
- *Game value:* The value of a terminal that will be reached assuming optimal strategies from both players (*minimax* value)
- *Search*: Find move that maximizes game value from current state



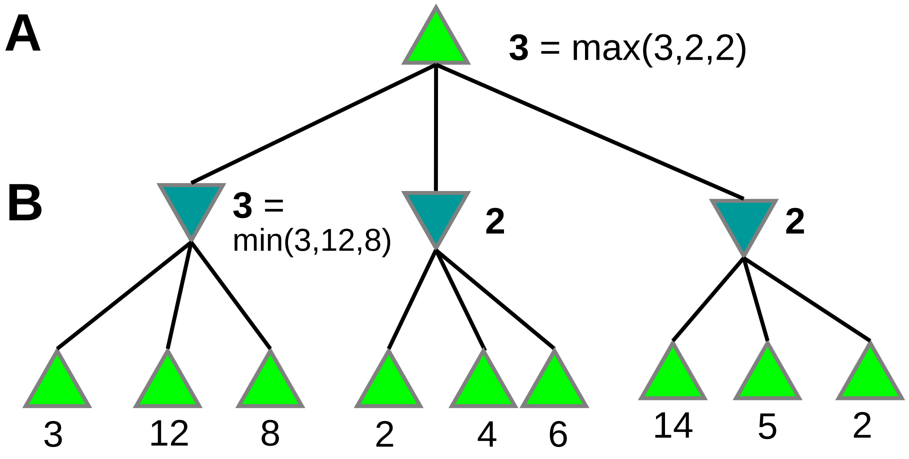
Optimal (minimax) Strategies

- Search the game tree such that:
  - A's turn to move → find the move that yields maximum payoff from the corresponding subtree → This is the move most favorable to A
  - B's turn to move → find the move that yields minimum payoff (best for B) from the corresponding subtree → This is the move most favorable to B

Minimax

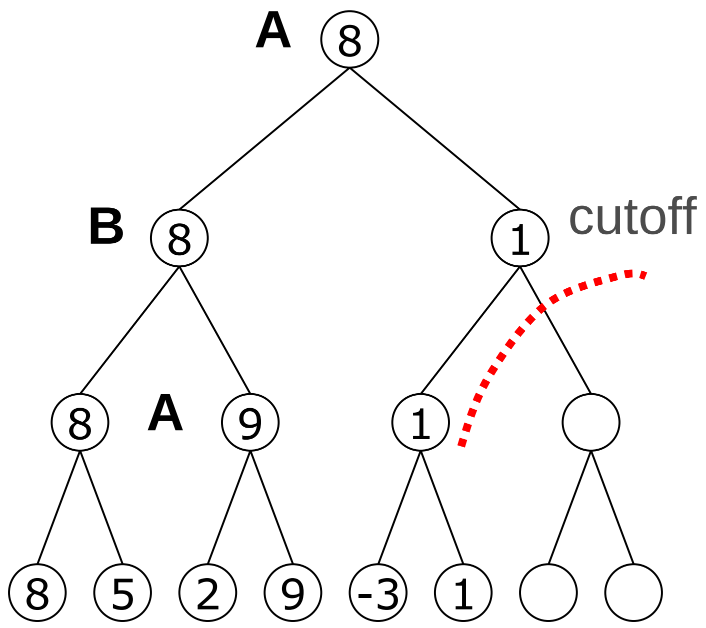
```
Minimax (s)
    If s is terminal
        Return U(s)

    If next move is A
        Return $\max_{s' \in Succs(s)} \text{Minimax}(s')$
    Else
        Return $\min_{s' \in Succs(s)} \text{Minimax}(s')$
```

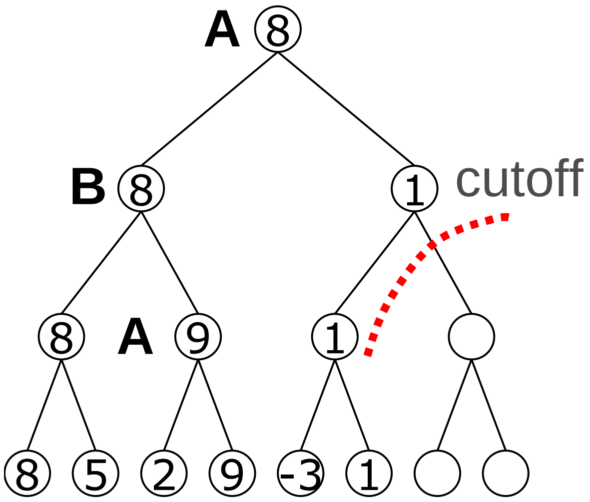


Minimax Properties

- *Complete:* If finite game
- *Optimal:* If opponent plays optimally
- *Complexity:* Essentially DFS, so:
  - – Time: O(B^m)
  - – Space: O(Bm)
  - – B = number of possible moves from any state (branching factor)
  - – m = depth of search (length of game)

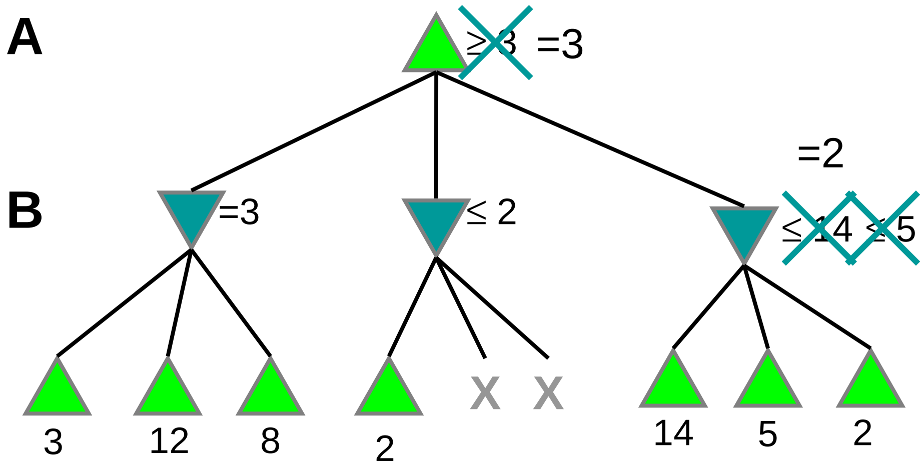


Pruning



- The value at A move is 8 (so far)
- If A moves right, the value there is 1 (so far)
- B will *never increase* the value at this node; it will always be less than 8
- B can *ignore* the remaining nodes

Pruning



αβ Pruning

- Maintain:
  - α = Best value found so far at A nodes, including those at current node
  - β = Best value found so far at B nodes, including those at current node
- If at a B node: No need to expand this node any further if α >= β because there is no way that a descendant of the current node can yield a better value

---

```
Minimax ($s, \alpha, \beta$)
    If $s$ is terminal
        Return $U(s)$
    If $A$ node
        For each $s'$ in $Succs(s)$
            $\alpha$ = Max($\alpha$, Minimax($s', \alpha, \beta$))
            If ($\alpha >= \beta$) Return $\beta$
        Return $\alpha$
    If $B$ node
        For each $s'$ in $Succs(s)$
            $\beta$ = Min($\beta$, Minimax($s', \alpha, \beta$))
            If ($\alpha >= \beta$) Return $\alpha$
        Return $\beta$
```

```
Minimax ($s, \alpha, \beta$)
    If $s$ is terminal
        Return $U(s)$
    If $A$ node
        For each $s'$ in $Succs(s)$
            $\alpha$ = Max($\alpha$, Minimax($s', \alpha, \beta$))
            If ($\alpha >= \beta$) Return $\beta$
        Return $\alpha$
    If $B$ node
        For each $s'$ in $Succs(s)$
            $\beta$ = Min($\beta$, Minimax($s', \alpha, \beta$))
            If ($\alpha >= \beta$) Return $\alpha$
        Return $\beta$
```

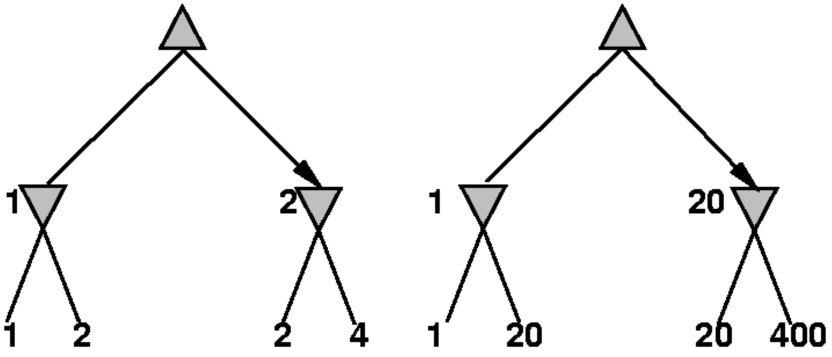
Same as default minimax

Prune if no better solution can be found

Properties

- Guaranteed to find same solution
- O($B^{m/2}$) with proper ordering of the nodes → At “A” node, the successor are in order from high to low score
- Use heuristic evaluation functions to cut off search early
- Example: Weighted sum of number of pieces (material value of state)
- Stop search based on cutoff test (e.g., maximum depth)
- Iterative deepening search to limit DFS
- Solve by brute-force dynamic programming when the number of states is small

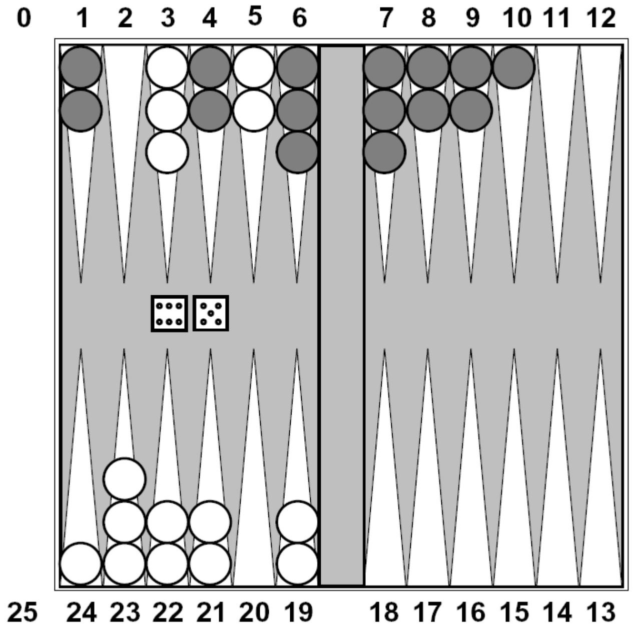
Choice of Value?



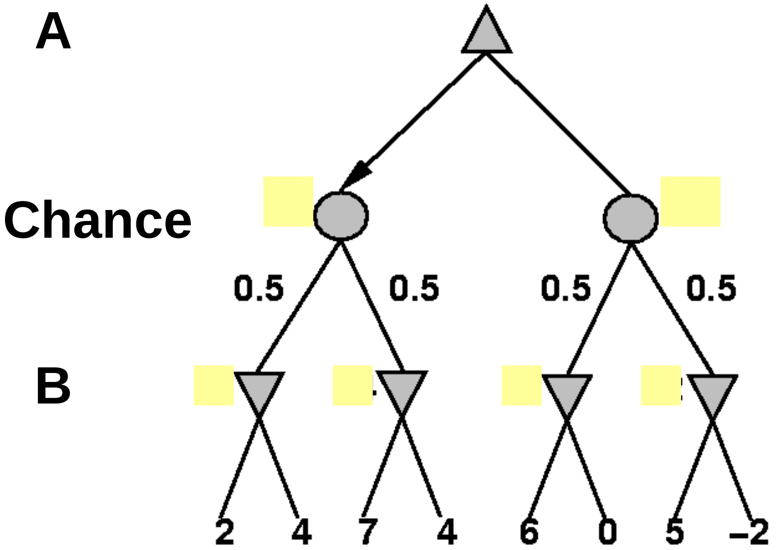
- Absolute game value is different in the two cases
- Minimax solution is the same
- Only the relative ordering of values matters, not the absolute values → *ordinal utility values*
- True only for *deterministic* games
- Evaluation functions can be any function that preserves the ordering of the utility values

---

Non-Deterministic Games



Non-Deterministic Games

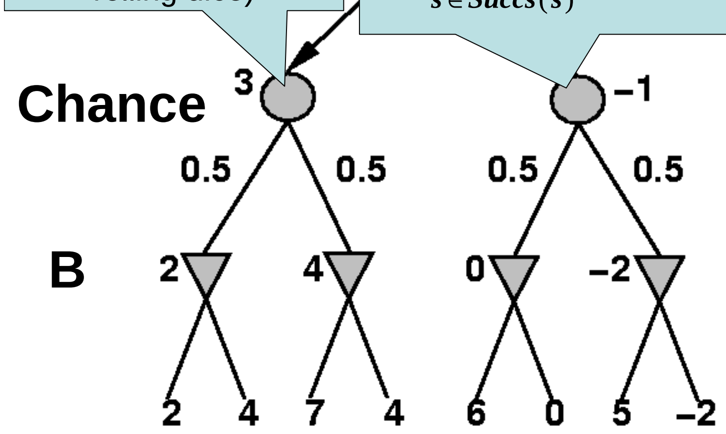


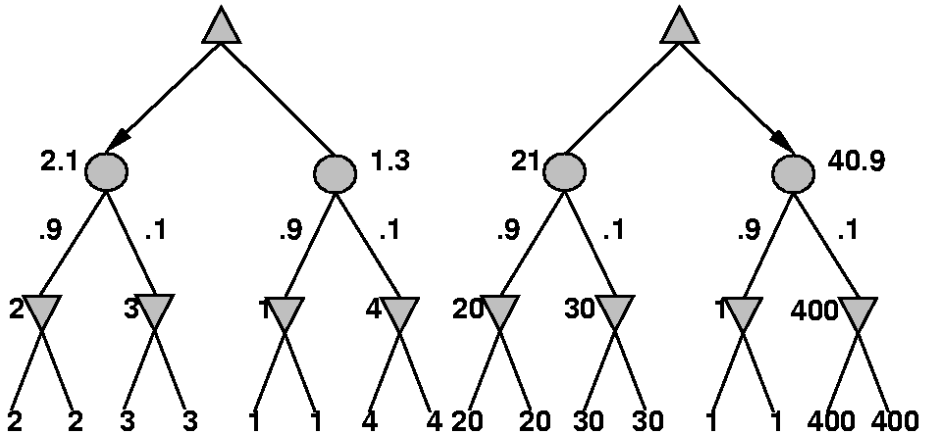
Non-Deterministic Games

Includes states where neither player makes a choice. A random decision is made (e.g., rolling dice)

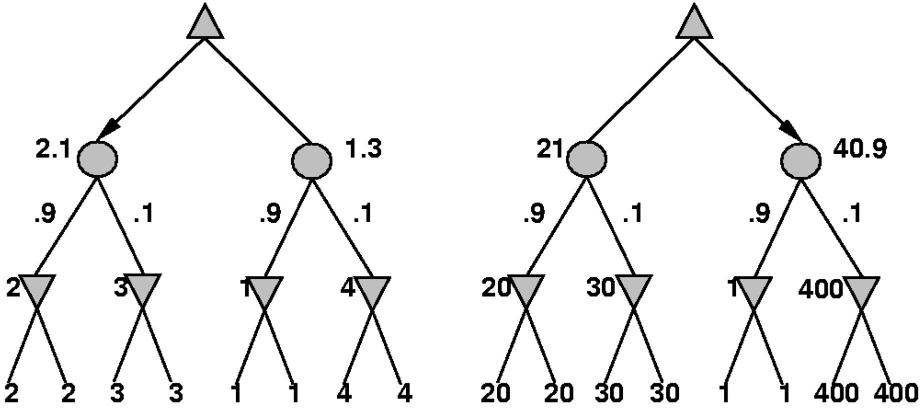
Use expected value of successors at chance nodes:

$$\sum_{s' \in Succs(s)} p(s') MiniMax(s')$$





Choice of Utility Values



- Different utility values may yield radically different result even though the order is the same → Absolute utility values do matter
- Utility should be proportional to actual payoff, it is not sufficient to follow the same order
- Think of choosing between 2 lotteries with same odds but radically different payoff distributions
- Implication: Evaluation functions must be *linear positive* functions of utility
- Kind of obvious but important consideration for later developments

Non-Deterministic Minimax

Minimax (*s*)

If *s* is terminal

Return *U*(*s*)

If next move is *A:* Return $\max_{s' \in Succs(s)} \text{Minimax}(s')$

If next move is *B* Return $\min_{s' \in Succs(s)} \text{Minimax}(s')$

If chance node Return $\sum_{s' \in Succs(s)} p(s') \text{Minimax}(s')$

Properties

- α-β pruning can be extended provided that the utility values are bounded → We don’t need to evaluate all the children of a chance node to bound the average
- Less effective
- Different outcomes depending on exact values of utility, not just ordering

Success stories

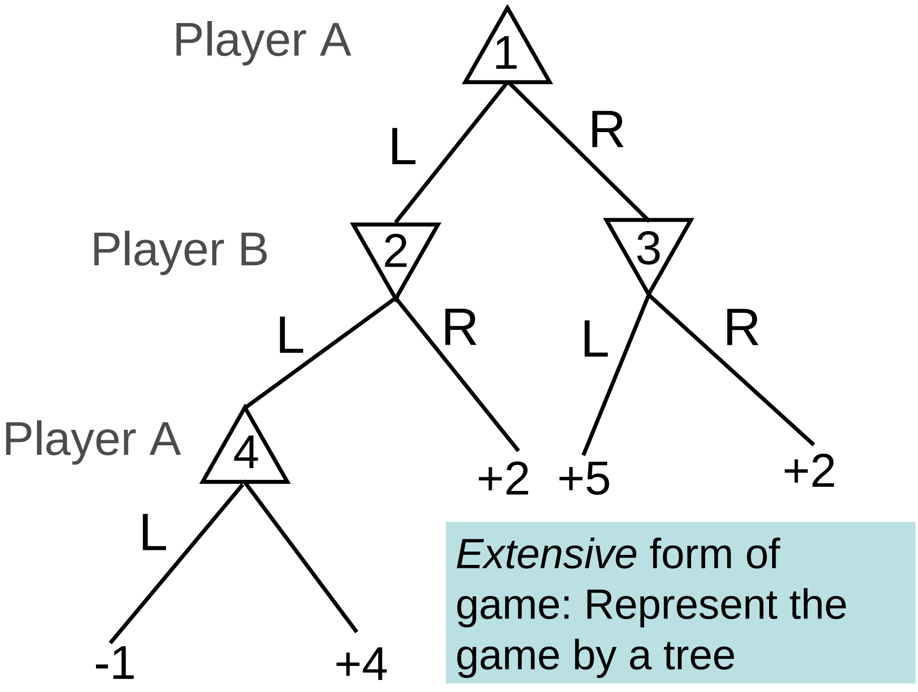
- Chess: Deep Blue
- Backgammon: TD-Backgammon
- Checkers
- Any more?
- What strategies would/wouldn't work?

Games with Hidden Information

R&N Chapter 6

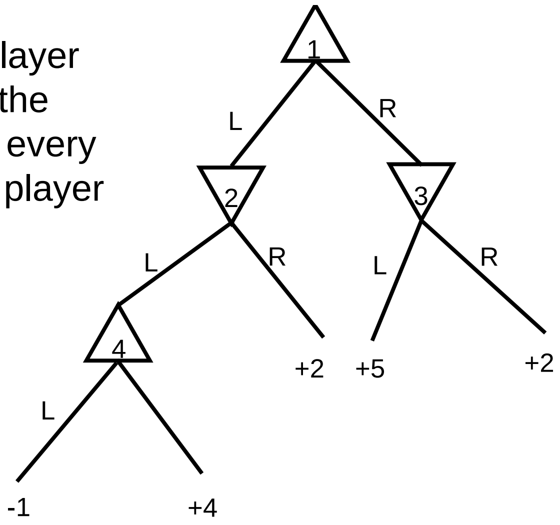
R&N Section 17.6

- Assumptions so far:
  - *Two-player game*: Player A and B.
  - *Perfect information*: Both players see all the states and decisions. Each decision is made *sequentially*.
  - *Zero-sum*: Player's A gain is exactly equal to player B's loss.
- We will eliminate the assumption of "perfect information" leading to far more realistic models.
  - Some more game-theoretic definitions → Matrix games
  - Minimax results for perfect information games
  - Minimax results for hidden information games

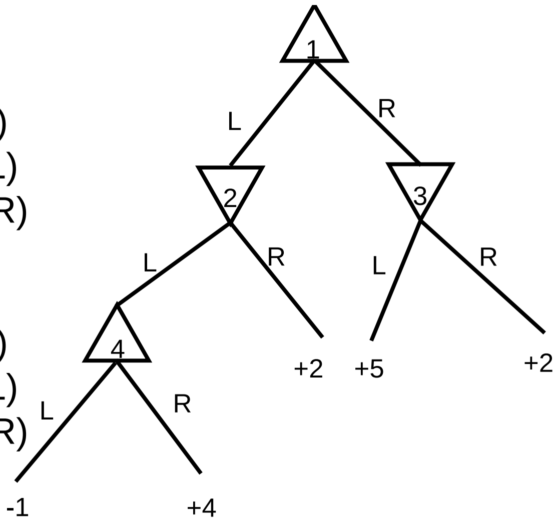


Extensive form of game: Represent the game by a tree

A *pure strategy* for a player defines the move that the player would make for every possible state that the player would see.



Pure strategies for A:
Strategy I: (1→L,4→L)
Strategy II: (1→L,4→R)
Strategy III: (1→R,4→L)
Strategy IV: (1→R,4→R)
Pure strategies for B:
Strategy I: (2→L,3→L)
Strategy II: (2→L,3→R)
Strategy III: (2→R,3→L)
Strategy IV: (2→R,3→R)

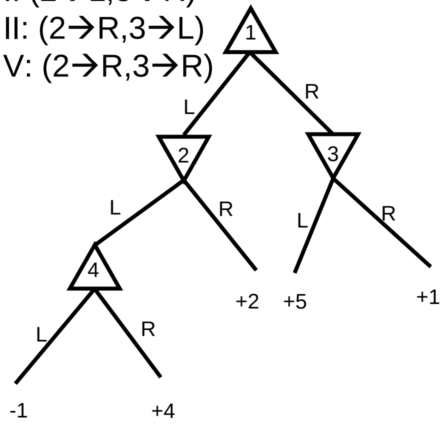


In general: If *N* states and *B* moves, how many pure strategies exist?

Matrix form of games

Pure strategies for A:
Strategy I: (1→L,4→L)
Strategy II: (1→L,4→R)
Strategy III: (1→R,4→L)
Strategy IV: (1→R,4→R)

Pure strategies for B:
Strategy I: (2→L,3→L)
Strategy II: (2→L,3→R)
Strategy III: (2→R,3→L)
Strategy IV: (2→R,3→R)



	I	**II**	**III**	**IV**
I	**-1**	**-1**	**+2**	**+2**
II	**+4**	**+4**	**+2**	**+2**
III	**+5**	**+1**	**+5**	**+1**
IV	**+5**	**+1**	**+5**	**+1**

Pure strategies for Player B

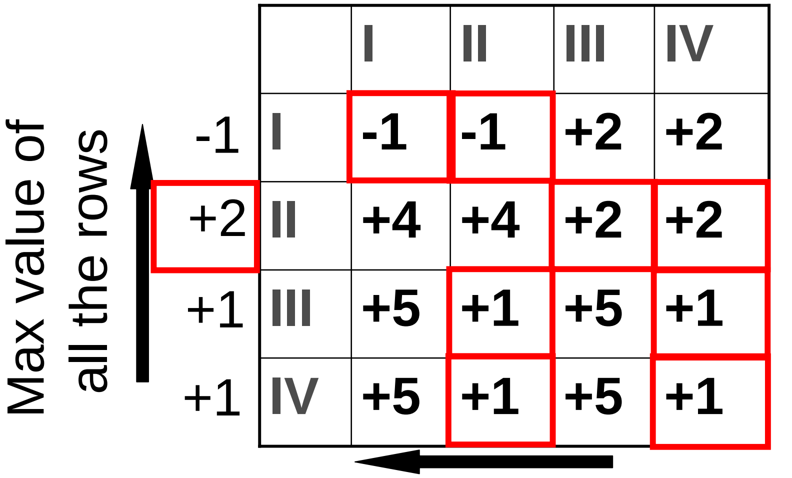
Pure strategies for Player A

	I	**II**	**III**	**IV**
I	**-1**	**-1**	**+2**	**+2**
II	**+4**	**+4**	**+2**	**+2**
III	**+5**	**+1**	**+5**	**+1**
IV	**+5**	**+1**	**+5**	**+1**

Player A's payoff if game is played with strategy I by Player A and strategy III by Player B

- *Matrix normal form* of games: The table contains the payoffs for all the possible combinations of pure strategies for Player A and Player B
- The table characterizes the game completely, there is no need for any additional information about rules, etc.
- Although, in many cases, the number of pure strategies may be too large for the table to be represented explicitly, the matrix representation is the basic representation that is used for deriving fundamental properties of games.

Minimax →Matrix version



Max value of all the rows

		I	II	III	IV
-1	I	-1	-1	+2	+2
+2	II	+4	+4	+2	+2
+1	III	+5	+1	+5	+1
+1	IV	+5	+1	+5	+1

Min value across each row

$$\underset{Rows\ i}{Max}\ \underset{Columns\ j}{Min}\ \mathbf{M}(i,j)$$

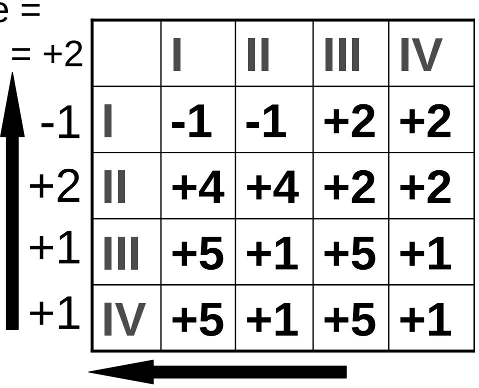
Minimax →Matrix version

- For each strategy (each row of the game matrix), Player A should assume that Player B will use the optimal strategy given Player A's strategy (the strategy with the minimum value in the row of the matrix). Therefore the best value that Player can achieve is the maximum over all the rows of the minimum values across each of the rows:

$$\underset{Rows\ i}{Max}\ \underset{Columns\ j}{Min}\ \mathbf{M}(i,j)$$

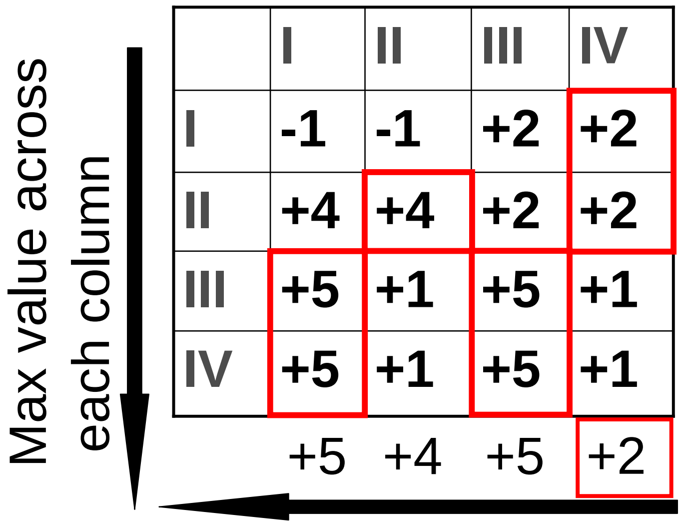
- The corresponding pure strategy is the optimal solution for this game → It is the optimal strategy for A assuming that B plays optimally.

Max value =
game value = +2



		I	II	III	IV
-1	I	-1	-1	+2	+2
+2	II	+4	+4	+2	+2
+1	III	+5	+1	+5	+1
+1	IV	+5	+1	+5	+1

Min value across each row



Max value across each column

	I	II	III	IV
I	-1	-1	+2	+2
II	+4	+4	+2	+2
III	+5	+1	+5	+1
IV	+5	+1	+5	+1
	+5	+4	+5	+2

Min of all the columns

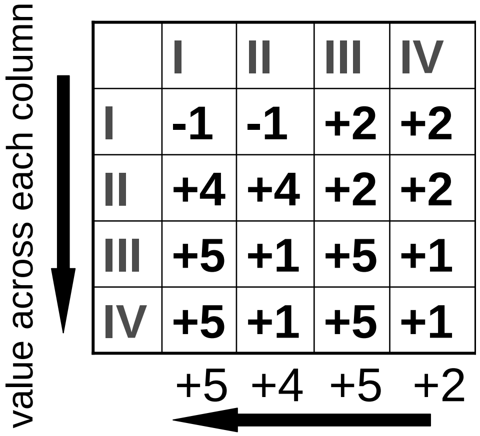
$$\underset{Columns\ j}{Min}\ \underset{Rows\ i}{Max}\ \mathbf{M}(i,j)$$

Minimax or Maximin?

- But we could have used the opposite argument:
- For each strategy (each column of the game matrix), Player B should assume that Player A will use the optimal strategy given Player B's strategy (the strategy with the maximum value in the column of the matrix):

$$\underset{Columns\ j}{Min}\ \underset{Rows\ i}{Max}\ \mathbf{M}(i,j)$$

- Therefore the best value that Player B can achieve is the minimum over all the columns of the maximum values across each of the columns
- Problem: Do we get to the same result??
- Is there always a solution?



Max value across each column

	I	II	III	IV
I	-1	-1	+2	+2
II	+4	+4	+2	+2
III	+5	+1	+5	+1
IV	+5	+1	+5	+1
	+5	+4	+5	+2

Min value =
game value = +2

Max value =
game value = +2

Note that we find the same value and same strategies in both cases. Is that always the case?

		I	II	III	IV
-1	I	-1	-1	+2	+2
+2	II	+4	+4	+2	**+2**
+1	III	+5	+1	+5	+1
+1	IV	+5	+1	+5	+1

Min value across each row

$$\underset{\text{Rows } i}{Max}\ \underset{\text{Columns } j}{Min}\ \mathbf{M}(i, j)$$

Max value across each column

	I	II	III	IV
I	-1	-1	+2	+2
II	+4	+4	+2	**+2**
III	+5	+1	+5	+1
IV	+5	+1	+5	+1
	+5	+4	+5	+2

Min value =
game value = +2

$$\underset{\text{Columns } j}{Min}\ \underset{\text{Rows } i}{Max}\ \mathbf{M}(i, j)$$

Minimax vs. Maximin

- Fundamental Theorem I (von Neumann):
  - For a two-player, zero-sum game with perfect information:
    - *There always exists an optimal pure strategy for each player*
    - *Minimax = Maximin*
- *Note: This is a game-theoretic formalization of the minimax search algorithm that we studied earlier.*

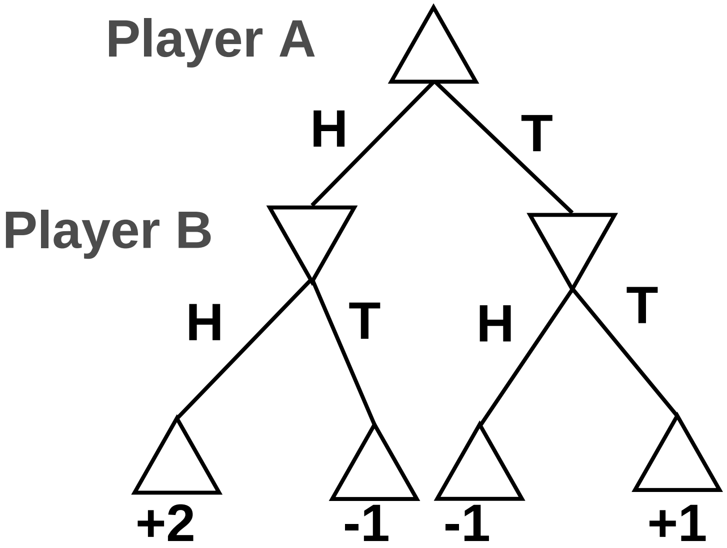
Another (Seemingly Simple) Game

- The two Players A and B each have a coin
- They show each other their coin, choosing to show either head or tail.
- If they both choose head → Player B pays Player A $2
- If they both choose tail → Player B pays Player A $1
- If they choose different sides → Player A pays Player B $1

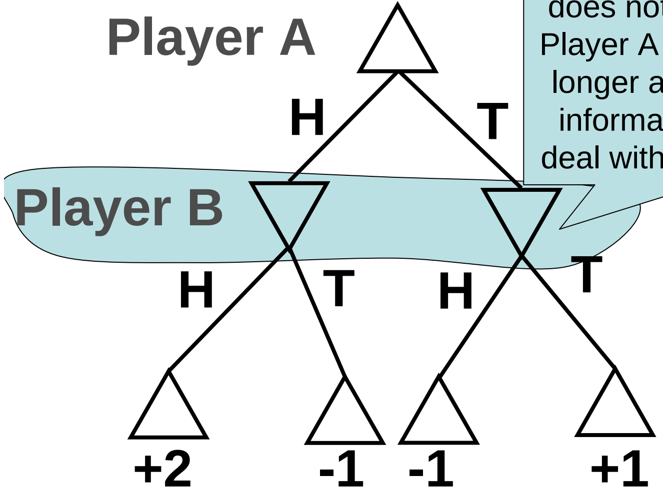
Side Note about all toy examples

- If you find this kind of toy example annoying, it models a large number of real-life situations.
- For example: Player A is a business owner (e.g., a restaurant, a plant..) and Player B is an inspector. The inspector picks a day to conduct the inspection; the owner picks a day to hide the bad stuff. Player B wins if the days are different; Player A wins if the days are the same.
- This class of problems can be reduced to the “coin game” (with different payoff distributions, perhaps).

Extensive Form



Extensive Form



Problem: Since the moves are simultaneous, Player B does not know which move Player A chose → This is no longer a game with perfect information → we have to deal with *hidden information*

Player B

Player A

	H	T
H	**+2**	**-1**
T	**-1**	**+1**

Matrix Normal Form

Player B

Player A

	H	T
H	**+2**	**-1**
T	**-1**	**+1**

- It is no longer the case that maximin = minimax (easy to verify: -1 vs. +1)
- Therefore: It appears that there is no pure strategy solution
- In fact, in general, *none of the pure strategies* are solutions to a zero-sum game with *hidden information*

Why no Pure Strategy Solutions?

Player B

Player A

	H	**T**
H	**+2**	**-1**
T	**-1**	**+1**

- Intuition:
- If Player A considers move H, he has to assume that Player B will choose the worst-case move (for A), which is move T
  - Therefore Player A should try move T instead, but then he has to assume that Player B will choose the worst-case move (for A), which is move H.
    - Therefore Player A should consider move H, but then he has to assume that Player B will choose the worst-case move (for A), which is move T.
      - Therefore Player A should try move T instead, but then he has to assume that [illegible] (for A), which is move H.
        - » Therefo[illegible]hen he has to assume that [illegible] B will choose the worst-case move (for A), which [illegible] move T.
        - » …………………………..

Help! Get me outta here!

Mixed Strategies

- It is no longer possible to find an optimal *pure* strategy for Player A.
- We need to change the problem a bit: We assume that Player A chooses a pure strategy *randomly* at the beginning of the game.
- In that scenario, Player A selects one pure strategy probability *p* and the other one with probability 1-*p*.
- This strategy of choosing pure strategies randomly is called a *mixed strategy* for Player A and is entirely defined by the probability *p*.
- Question: We know that we cannot find an optimal pure strategy for Player A, but can we find an optimal mixed strategy *p*?
- Answer: Yes! The result that we derived for the simple example holds for general games. It yields a procedure for finding the optimal mixed strategy for zero-sum games.

Summary

- Definitions
- Game evaluation
- Optimal solutions
  - Minimax
  - Alpha-beta pruning
- Approximations
  - Heuristic evaluation functions
  - Cutoffs
  - Endgames
- Non-deterministic games

Summary

- Matrix form of games
- Minimax procedure and theorem for games with perfect information → Always a *pure strategy* solution
- Minimax procedure and theorem for games with hidden information → Need for a *mixed strategy* solution