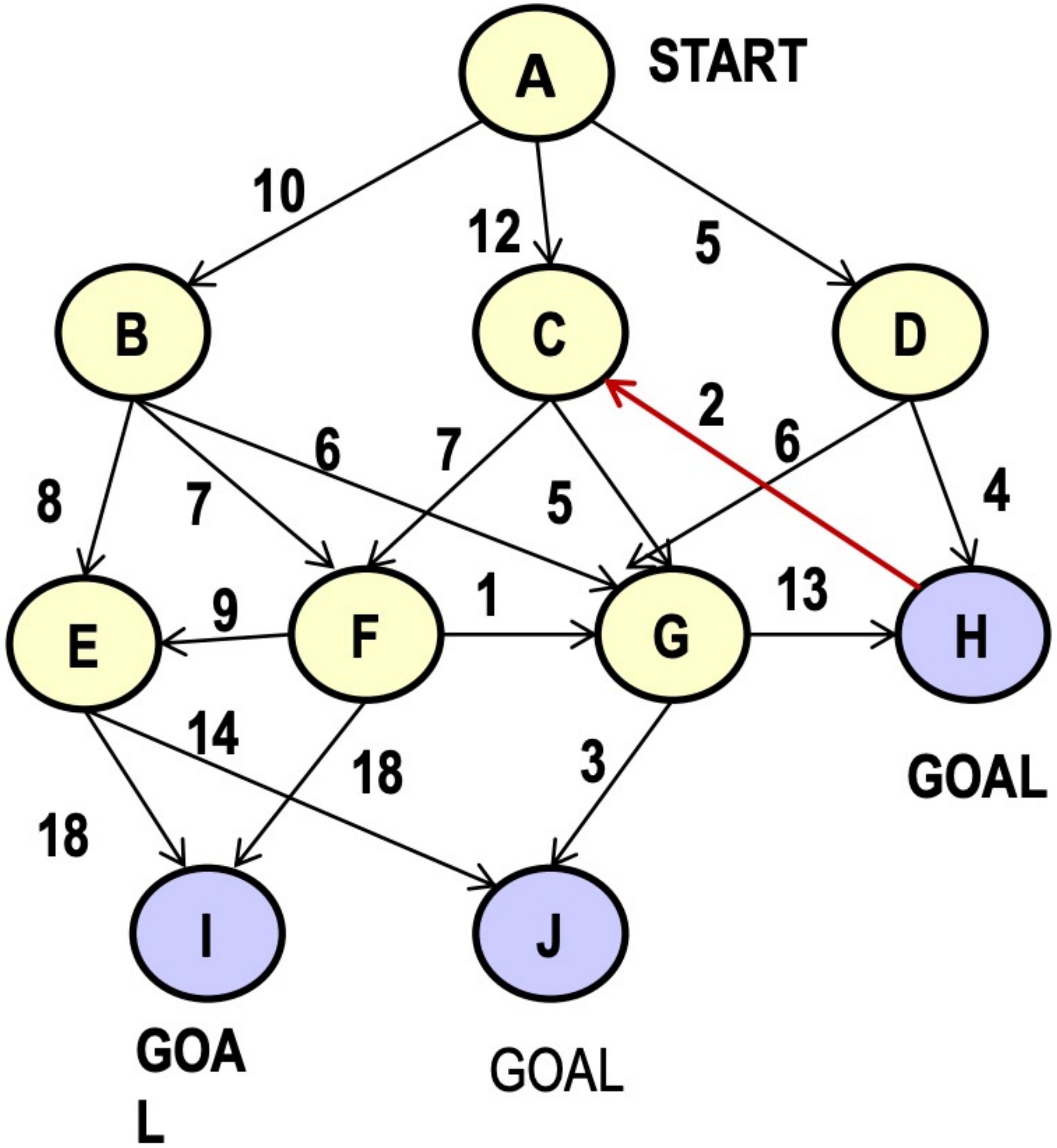
# ARTIFICIAL INTELLIGENCE (AI)
# STATE SPACE AND SEARCH ALGORITHMS

Shreya Ghosh
Assistant Professor, Computer Science and Engineering
*IIT Bhubaneswar*

# SEARCHING STATE SPACE GRAPHS WITH EDGE COSTS



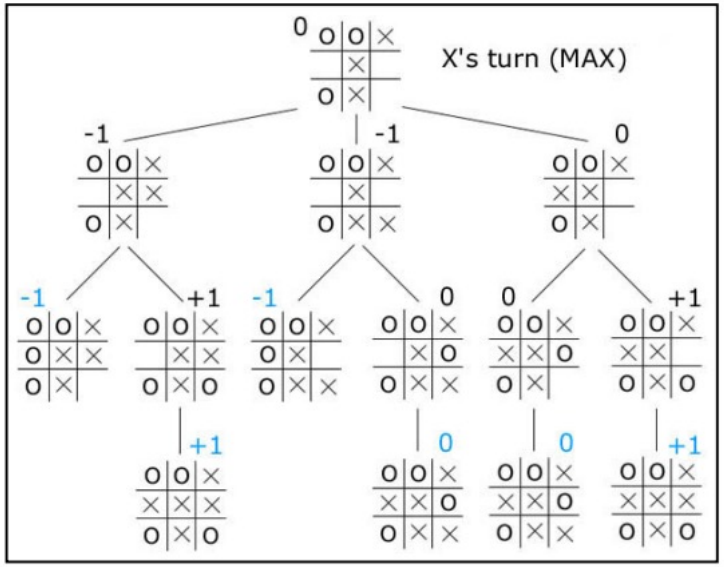
# ALGORITHM IDA*

## ITERATIVE DEEPENING A* (IDA*)

1. Set Cut-off Bound to f(s)
2. Perform DFBB with Cut-off Bound. Backtrack from any node whose f(n) > Cut-off Bound.
3. If Solution is Found, at the end of one Iteration, Terminate with Solution
4. If Solution is not found in any iteration, then update Cut-off Bound to the lowest f(n) among all nodes from which the algorithm Backtracked.
5. Go to Step 2
6. PROPERTIES OF DFBB AND IDA*: Solution Cost, Memory, Node expansions, Heuristic Accuracy, Performance on Trees / Graphs

# GAMES


File Edit View Insert Game Engine Tools Window Help
0:00:00
0:00:04


X's turn (MAX)

# MULTI-PLAYER GAMES

South Deals
N-S Vul

North:
♠ K J 8 5
♥ J 10 2
♦ J 8
♣ A J 10 4

West:
♠ 10 6
♥ A 6 4 3
♦ A K 10 5 2
♣ Q 2

N
W E
S

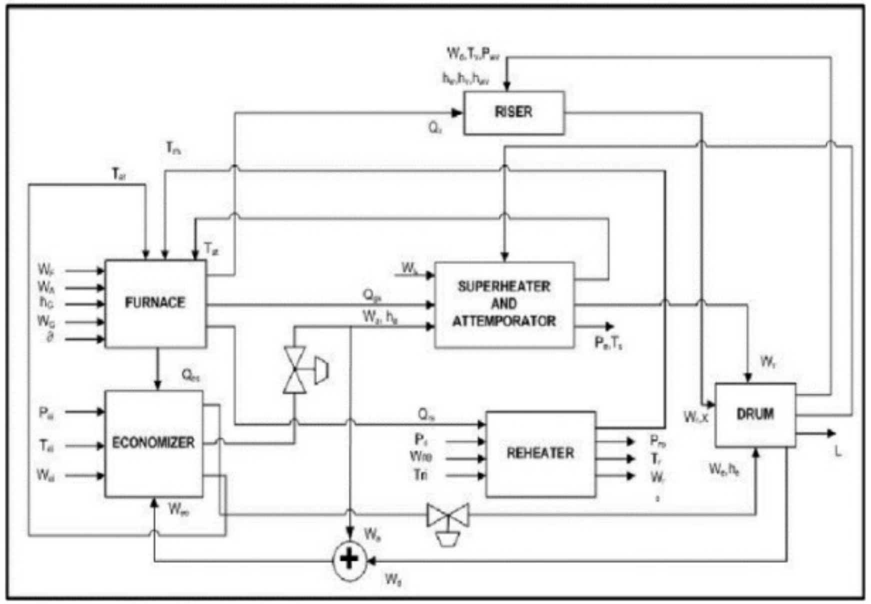
East:
♠ Q 9 2
♥ 8 7 5
♦ Q 9 6 3
♣ K 8 6

South:
♠ A 7 4 3
♥ K Q 9
♦ 7 4
♣ 9 7 5 3

| *West* | *North* | *East* | *South* |
|---|---|---|---|
| | | | Pass |
| 1 ♦ | Dbl | 2 ♦ | 2 ♠ |
| 3 ♦ | 3 ♠ | All pass | |

# PROBABILISTIC GAMES

BLOCK DIAGRAM OF A BOILER SYSTEM



# ROBOT GAMES

# PRISONER'S DILEMMA

- Two members of a criminal gang are arrested and imprisoned. Each prisoner is in solitary confinement with no means of communicating with the other.
- The prosecutors lack sufficient evidence to convict the pair on the principal charge, but they have enough to convict both on a lesser charge.
- Simultaneously, the prosecutors offer each prisoner a bargain

# PRISONER'S DILEMMA

| | | PRISONER B | |
|---|---|---|---|
| | | Prisoner B stays silent (cooperates) | Prisoner B betrays (defects) |
| PRISONER A | Prisoner A stays silent (cooperates) | Each serves 1 year | Prisoner A: 3 yrs<br>Prisoner B: goes free |
| | Prisoner A betrays (defects) | Prisoner A: goes free<br>Prisoner B: 3 yrs | Each serves 2 yrs |

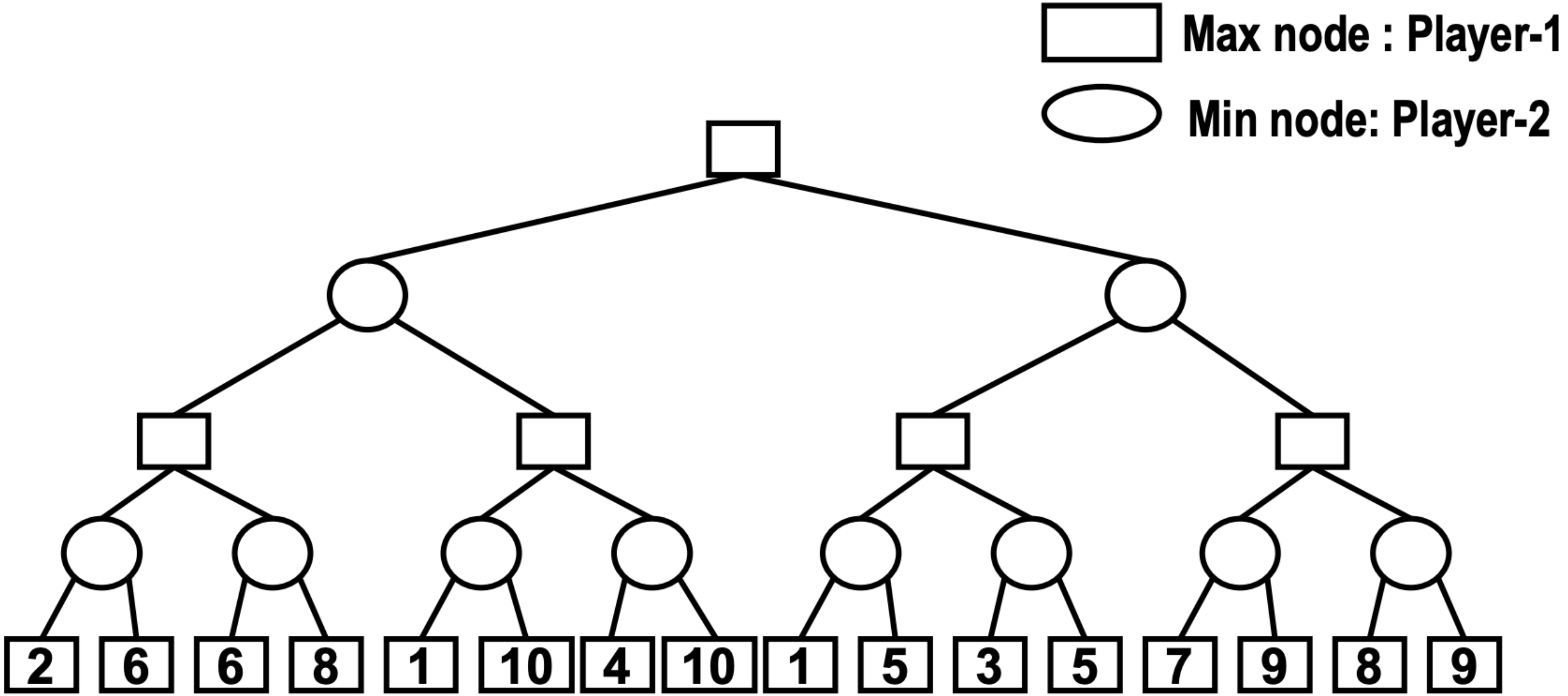
# GAME TREE

- A tree with three types of nodes, namely Terminal nodes, Min nodes and Max nodes.
- Terminal nodes have no children. The tree has alternating levels of Max and Min nodes, representing the turns of Player-1 and Player-2 in making moves
- All nodes represent some state of the game
- Terminal nodes are labeled with the payoff for Player-1. It could be Boolean (such as WON or LOST). In large games, where looking ahead up to the WON / LOST states is not feasible, the payoff at a terminal node may represent a heuristic cost representing the quality of the state of the game from Player-1's perspective

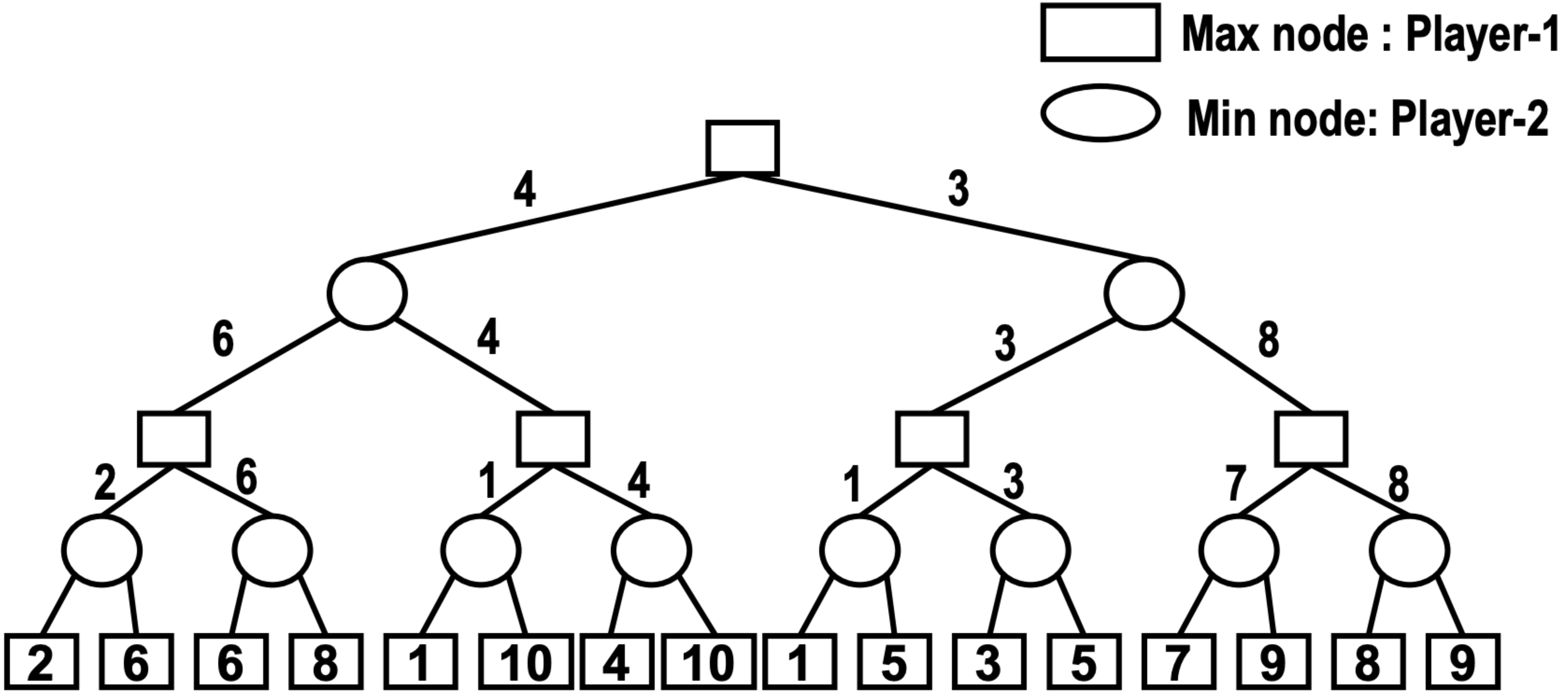
- The payoff at a Min node is the minimum among the payoffs of its successors
- The payoff at a Max node is the maximum among the payoffs of its successors

- If Player-1 aims to maximize its payoff, then it represents Max nodes, else it represents Min nodes

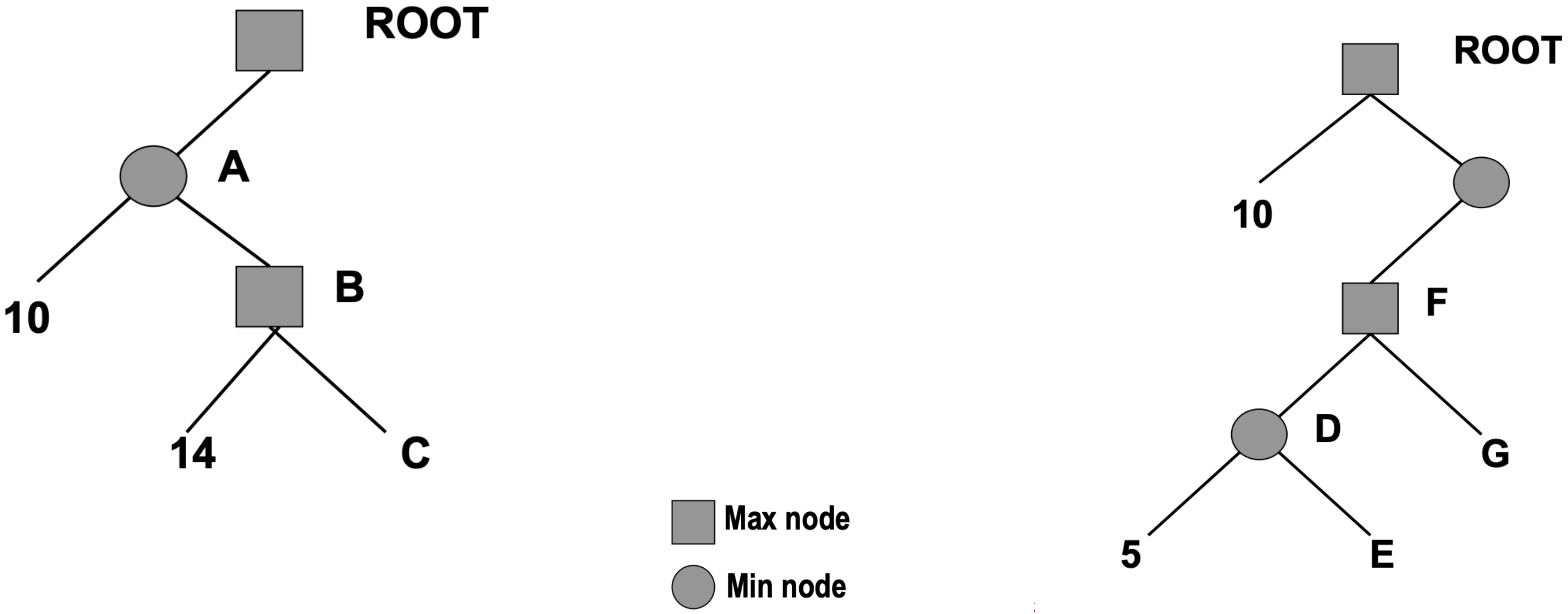
# SAMPLE GAME TREE



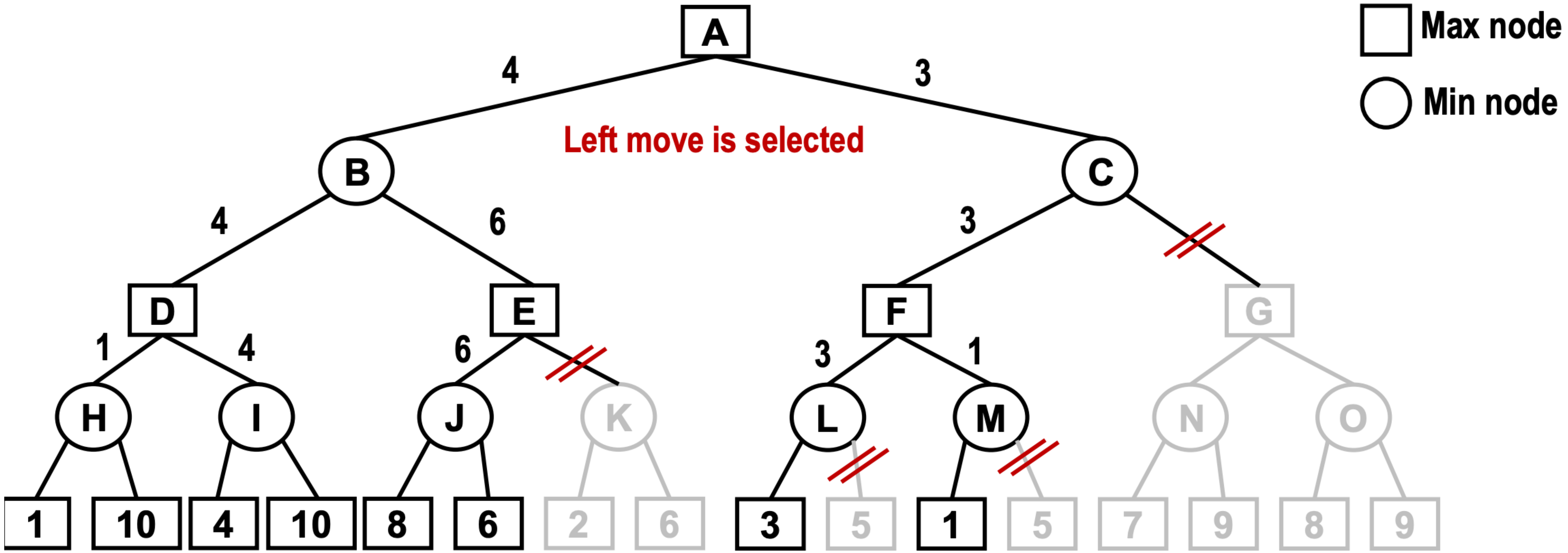
# GAME TREE: MINMAX VALUE



# Shallow and Deep Pruning



# Pruning explained



# ALPHA-BETA PRUNING IN GAME TREE SEARCH

- **Alpha Bound of J ($\alpha$) :**
  - The max current payoff of all MAX ancestors of J (Lower Bound)
  - Exploration of a min node, J, is stopped when its payoff $\beta$ (Upper Bound) equals or falls below alpha.
- **Beta Bound of J ($\beta$):**
  - The min current payoff of all MIN ancestors of J (Upper Bound)
  - Exploration of a max node, J, is stopped when its payoff $\alpha$ (Lower Bound) equals or exceeds beta
- In a max node, we update alpha or Lower Bound
- In a min node, we update beta or Upper Bound
- In both min and max nodes, we return when $\alpha \geq \beta$

# ALPHA-BETA PRUNING PROCEDURE $V(J;\alpha,\beta)$

1. If J is a terminal, return $V(J) = h(J)$.
2. If J is a max node:

   For each successor $J_k$ of J in succession:

   Set $\alpha = \max \{ \alpha, V(J_k; \alpha, \beta) \}$

   If $\alpha \geq \beta$ then return $\beta$, else continue

   Return $\alpha$
3. If J is a min node:

   For each successor $J_k$ of J in succession:

   Set $\beta = \min \{ \beta, V(J_k; \alpha, \beta) \}$

   If $\alpha \geq \beta$ then return $\alpha$, else continue

   Return $\beta$

The initial call is with
$V(\text{Root}; -\infty, +\infty)$