

Unit-6: **Game Playing**

Computer Engineering Department



Outline

- Overview
- MiniMax Search Procedure
- Alpha-Beta Cut-offs
- Refinements Theory

Introduction

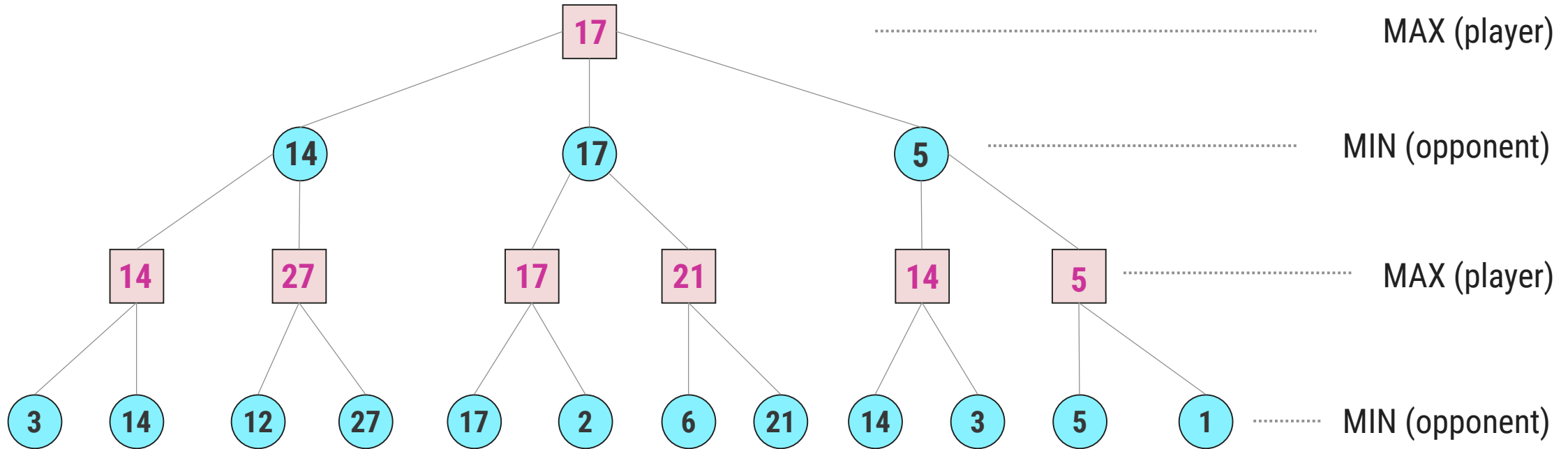
- ▶ Game Playing is an important domain of **Artificial Intelligence**.
- ▶ There are two reasons that games appeared to be a good domain.
 1. They provide a **structured task** in which it is very easy to measure success or failure.
 2. They are **easily solvable** by a straightforward search from the starting state to a winning position.
- ▶ Games require only the **domain knowledge** such as the rules, legal moves and the conditions of winning or losing the game.
- ▶ In a two-player game, both the players try to win the game. So, both of them try to make the **best move possible** at each turn.
- ▶ **To improve the effectiveness of a search** based problem solving program two things can be done.
 1. Improve **generate procedure** so that only good moves are generated.
 2. Improve **test procedure** so that the best move will be recognized and explored first.

Introduction

- ▶ If we use **legal-move generator** then the test procedure will have to look at each of them, because the test procedure must look at so many possibilities and it must be fast.
- ▶ The depth of the resulting **tree or graph** and its branching factor will be too large.
- ▶ Instead of legal-move generator we can use **plausible-move generator** in which only some small numbers of promising moves are generated.
- ▶ As the number of legal available moves increases it becomes increasingly important in applying **heuristics** to select only those moves that seem more promising.
- ▶ The performance of overall system can be improved **by adding heuristic knowledge** into both the generator and the tester.
- ▶ It is possible to search tree only ten or twenty moves deep then in order to choose the best move, the resulting board positions **must be compared** to discover which is most advantageous.

Introduction

- ▶ This is done using **static evaluation function**, which uses whatever information it has to evaluate individual board position by estimating how likely they are to lead eventually to a win.
- ▶ The most common search technique in game playing is **Minimax search procedure**.



The MINIMAX Search Procedure

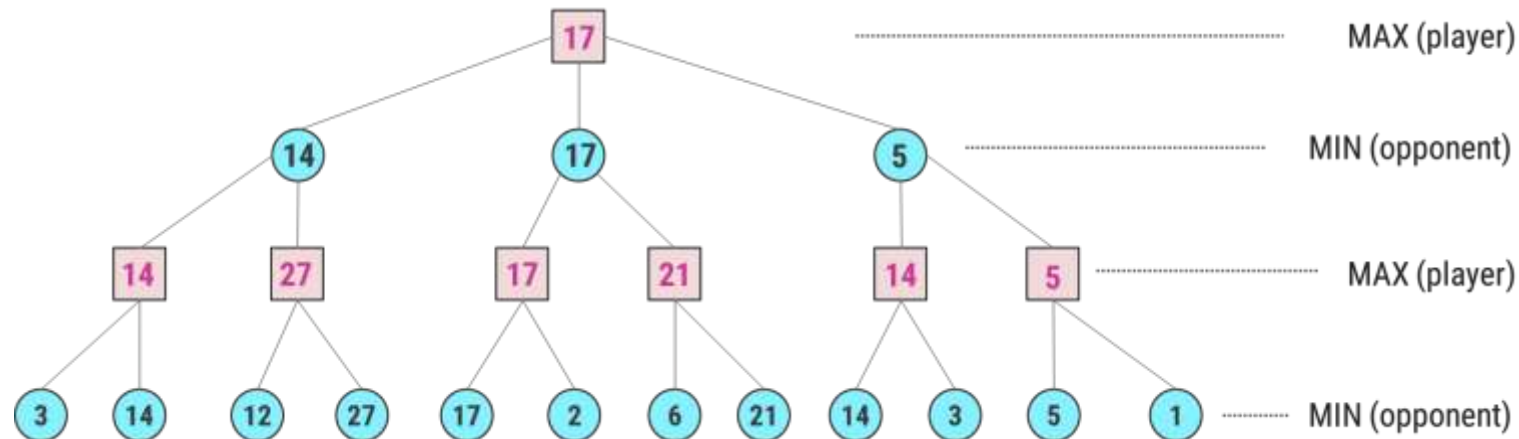
- ▶ The Minimax search is a **depth first and depth limited** procedure.
- ▶ The idea is to start at the current position and use the plausible-move generator to generate the **set of possible successor positions**.
- ▶ Now we can apply the **static evaluation function** to those positions and simply choose the best one.
- ▶ After doing so, we can back that value up to the starting position.
- ▶ Here, we assume that static evaluation function returns **larger values** to indicate good situations for us.
- ▶ So our goal is **to maximize the value** of the static evaluation function of the next board position.
- ▶ The opponents' goal is **to minimize the value** of the static evaluation function.

The MINIMAX Search Procedure

- ▶ The **alternation of maximizing and minimizing** at alternate ply when evaluations are to be pushed back up corresponds to the opposing strategies of the two players is called **MINIMAX**.
- ▶ It is recursive procedure that depends on two procedures :
 1. **MOVEGEN(position, player)**— The plausible-move generator, which returns a list of nodes representing the moves that can be made by Player in Position.
 2. **STATIC(position, player)** -- static evaluation function, which returns a number representing the goodness of Position from the standpoint of Player.
- ▶ To decide when recursive procedure should stop, variety of factors may influence the decision such as,
 - ↪ Has one side won?
 - ↪ How many ply have we already explored? Or how much time is left?
 - ↪ How stable is the configuration?

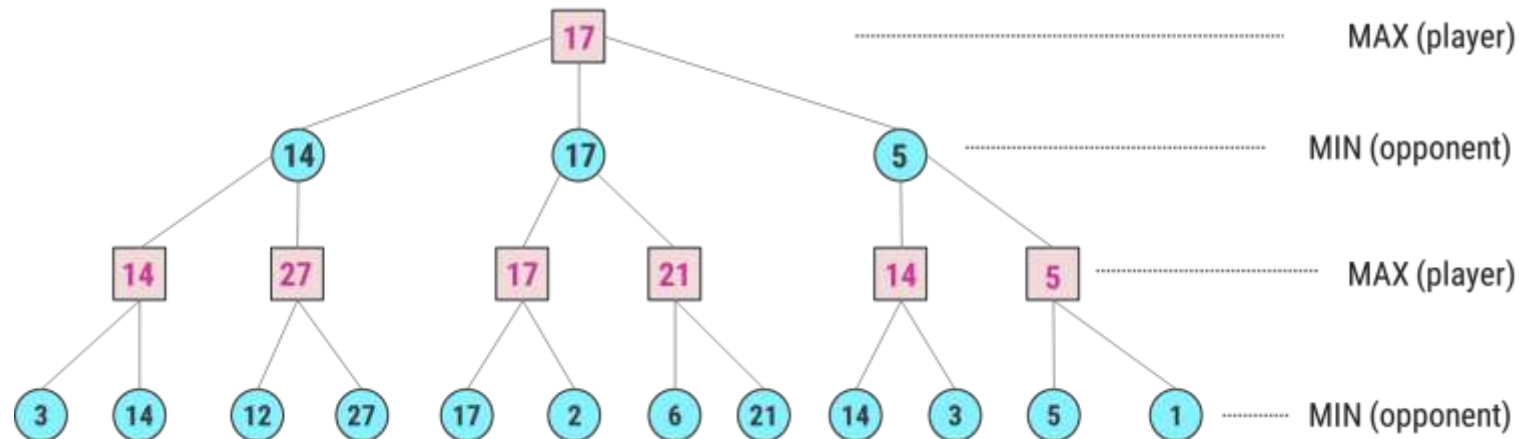
Minimax – Algorithm

- ▶ Given a game tree, the optimal strategy can be determined from the minimax value of each node, which we write as $\text{MINIMAX}(n)$.
- ▶ The minimax algorithm computes the **minimax decision** from the current state.
- ▶ It uses a **simple recursive computation** of the minimax values of each successor state, directly implementing the defining equations.
- ▶ The recursion proceeds all the way **down to the leaves** of the tree, and then the minimax values are **backed up through** the tree as the recursion unwinds.



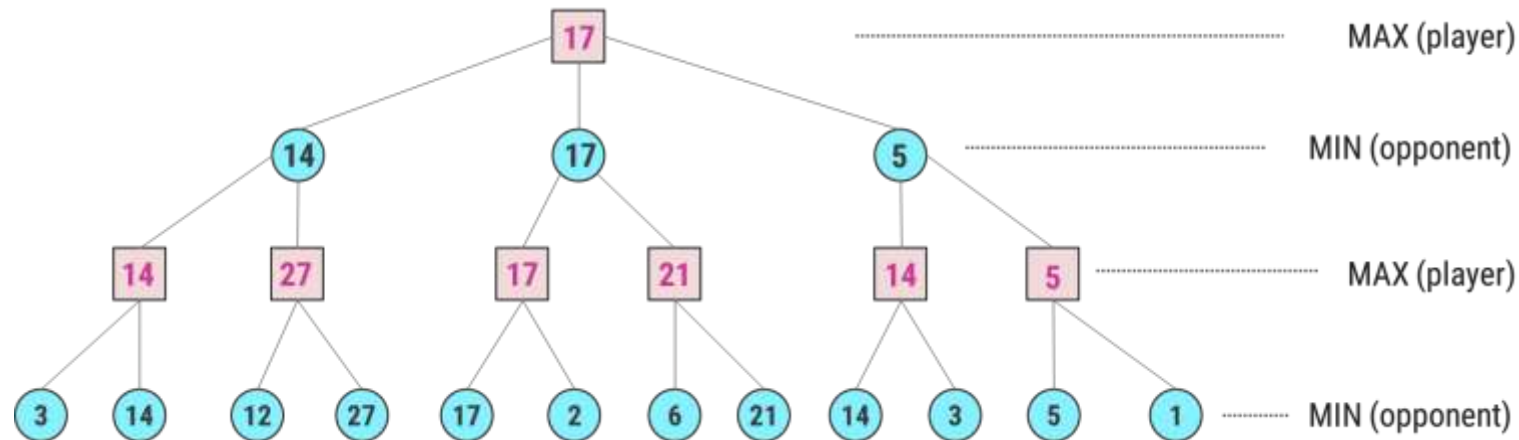
Minimax – Algorithm

- ▶ The minimax value of a node is the **utility (for MAX)** of being in the corresponding state, assuming that both players play optimally from there to the end of the game.
- ▶ The algorithm first recurses down to the three bottom left nodes and uses the **UTILITY function** on them to discover that their values are 3 and 14 respectively.
- ▶ Then it takes the maximum of these values, 14, and returns it as the backed up value of parent node.
- ▶ A similar process gives the backed-up values of 27, 17, 21, 14 and 5 respectively.



Minimax – Algorithm

- ▶ Then it takes the **minimum** of these values 14, 17 and 5 respectively.
- ▶ Finally, we take the **maximum** of 14, 17, and 5 to get the backed-up value of 17 for the root node.
- ▶ The minimax algorithm performs a complete **depth-first exploration** of the game tree.
- ▶ During every ply MAX prefers to move to a state of maximum value, whereas MIN prefers a state of minimum value.



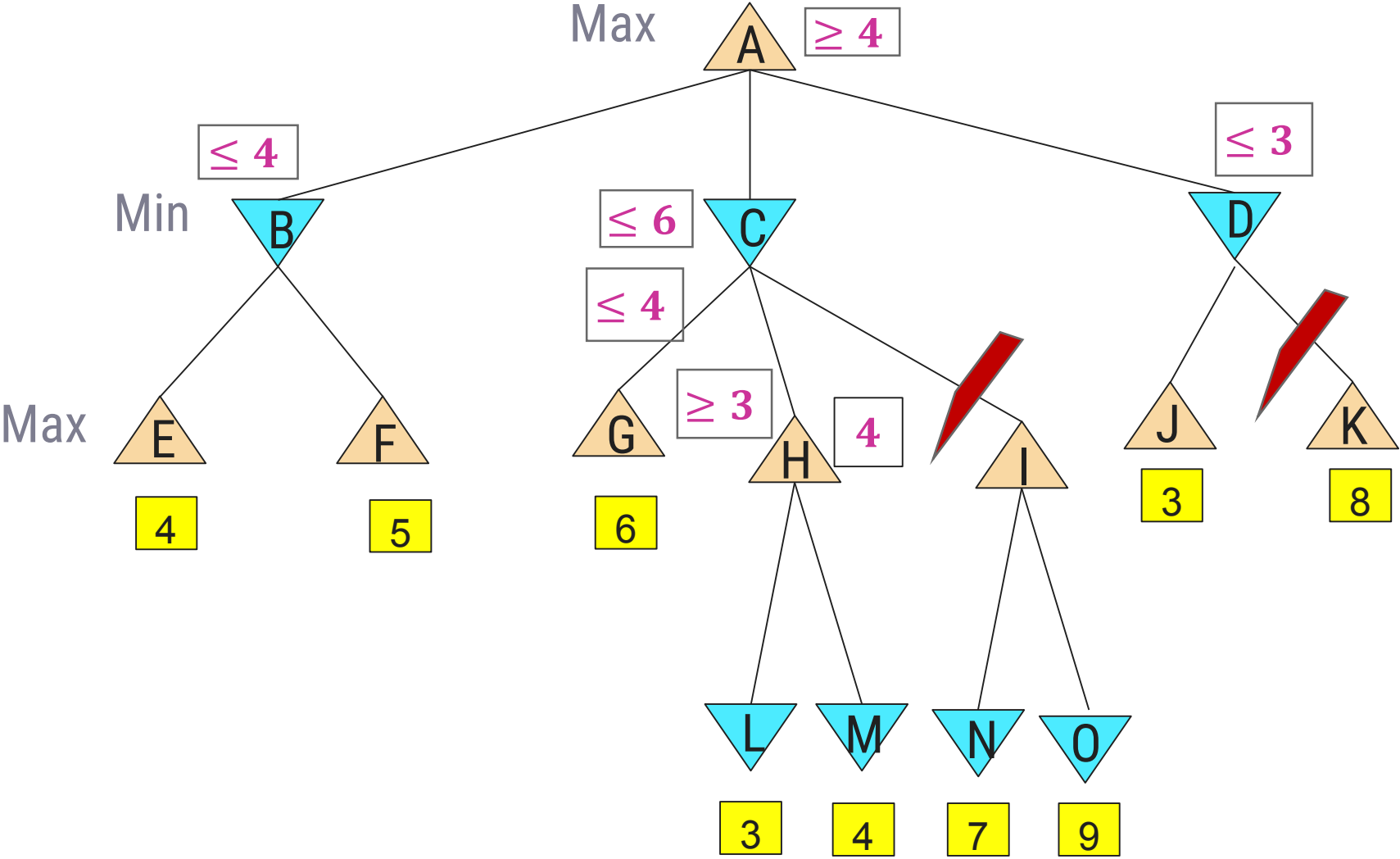
Alpha-Beta Pruning

- ▶ Alpha-beta pruning is a modified version of the Minimax algorithm. It is an optimization technique for the Minimax algorithm.
- ▶ In the Minimax search algorithm, the number of game states to be examined can be exponential in the depth of a tree.
- ▶ Hence there is a technique by which without checking each node of the game tree we can compute the correct Minimax decision, and this technique is called pruning.
- ▶ This involves two threshold parameter Alpha and beta for future expansion, so it is called alpha-beta pruning. It is also called as Alpha-Beta Algorithm.
- ▶ Alpha-beta pruning can be applied at any depth of a tree, and sometimes not only it prunes the tree leaves but also entire sub-tree.

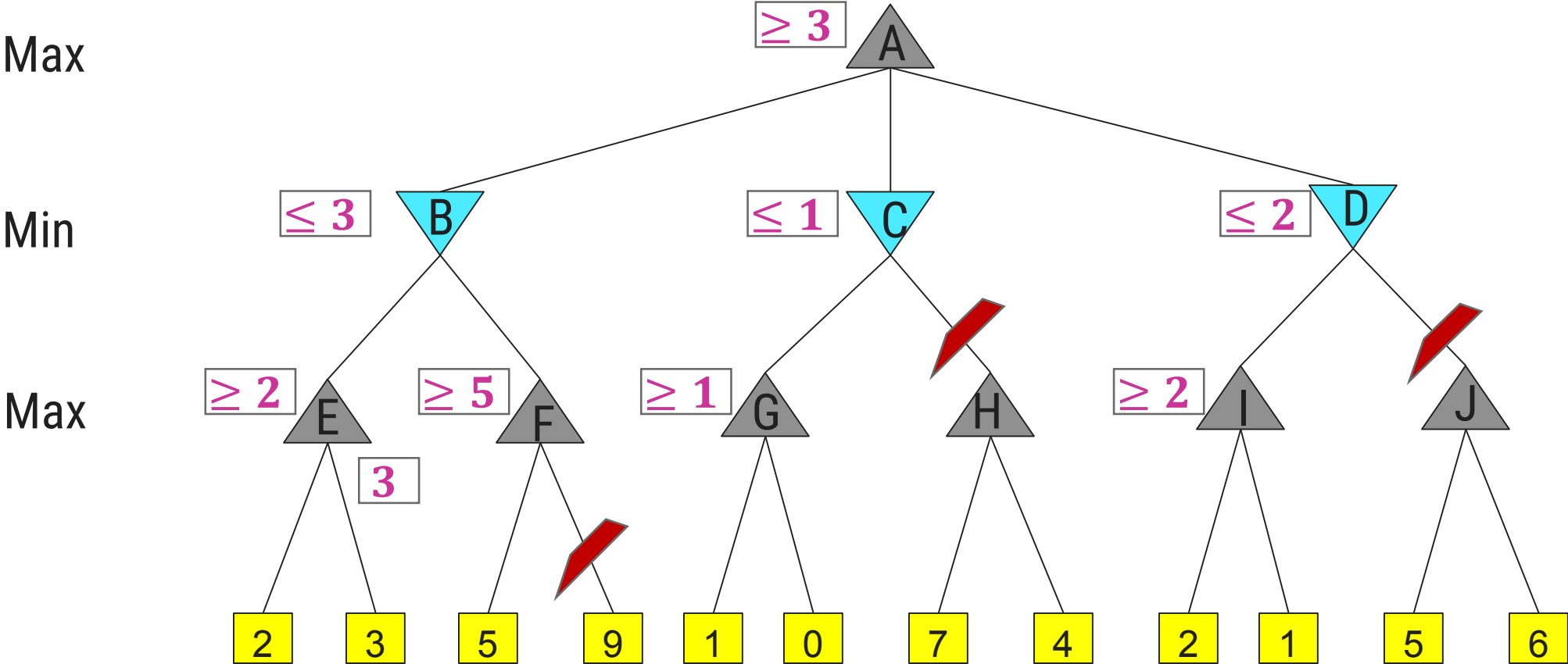
Alpha-Beta Pruning

- ▶ Alpha-beta pruning technique maintains two bounds:
 1. **Alpha (α)**: The best (highest-value) choice we have found so far at any point along the path of Maximizer. The initial value of alpha is $-\infty$. A lower bound on best, i.e., Max
 2. **Beta (β)**: The best (lowest-value) choice we have found so far at any point along the path of Minimizer. The initial value of beta is $+\infty$. An upper bound on what the opponent can achieve
- ▶ The Search proceeds maintaining α and β
- ▶ Whenever $\alpha \geq \beta$ *higher*, or $\beta \leq \alpha$ *higher*, searching further at this node is irrelevant.
- ▶ In conclusion, Minimax with alpha beta pruning is a faster algorithm than the Minimax algorithm.

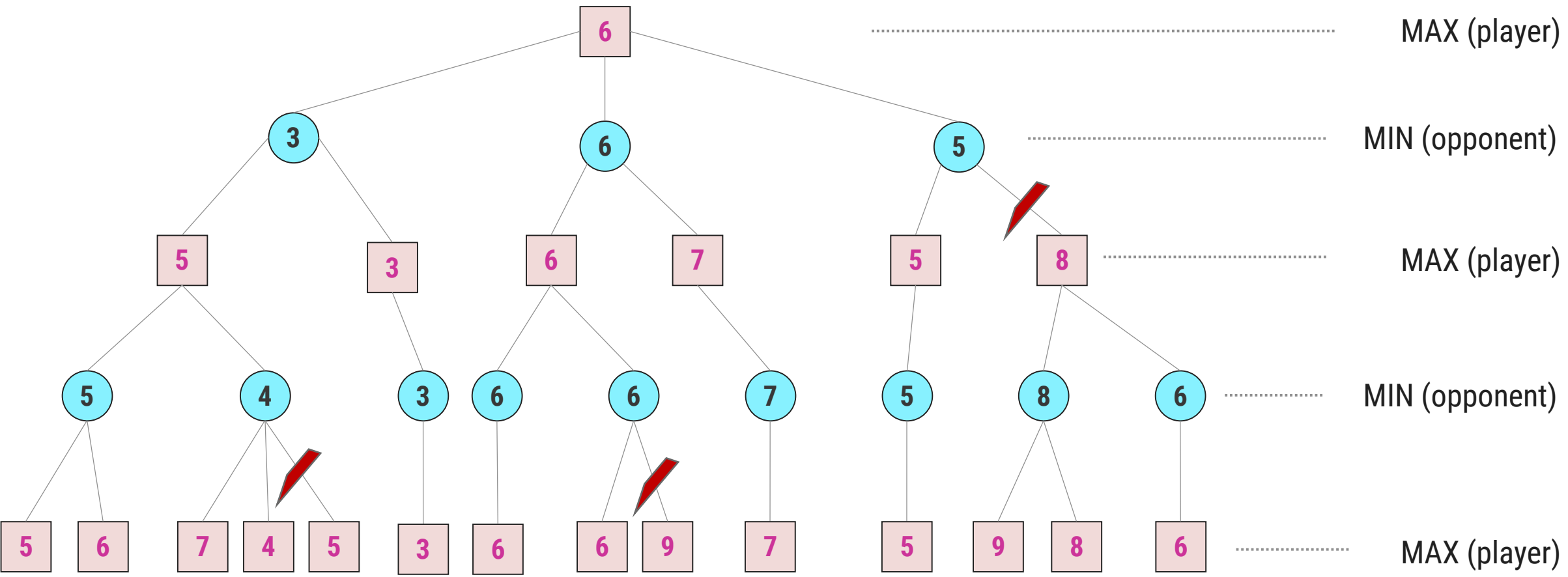
Minimax with Alpha-beta Pruning – Example 1



Minimax with Alpha-beta Pruning – Example 2



Minimax with Alpha-beta Pruning – Example 3



Game Refinement Theory

- ▶ Game theory is a discipline which stands from the game player's point of view with a focus on **how to win a game**.
- ▶ However, game designers would consider another important aspect: **how to make a game more attractive**.
- ▶ With such motivation, a new game theory from the game designer's point of view, called game refinement theory was proposed **in the early 2000s**.
- ▶ **Von Neumann** was a pioneer who formed the foundation for the modern game theory, which has widely been applied in various fields.
- ▶ One direction with game theory was to find **the best move** in a game or to ensure the possibility of winning the game based on the understanding of current positions.
- ▶ Another direction with game refinement theory was **to assess the attractiveness** or sophistication of a game.

Game Refinement Theory

- ▶ In particular, game refinement theory gives a measure to quantify the sophistication of a game.
- ▶ This enables to obtain the deep insight into the current game and improve the quality of the game.
- ▶ The measure of game refinement can also be used to obtain the deep insight into the history of games.
- ▶ It also gives a reasonable look on the evolution of specific game variants.
- ▶ Game refinement theory has been widely applied to many different types of games with the promising results.
- ▶ We can extend the idea of game refinement into the other domains in human life such as sports games, video games, education or business.
- ▶ In many activities of human, the engagement is used as one of the important standards to evaluate the effectiveness of those activities.

Thank You!