Artificial Intelligence

Engr. Madeha Mushtaq
Department of Computer Science
Iqra National University

Adversarial Search

- Until now all the searches that we have studied, there was only one person or agent searching the solution space to find the goal or the solution.
- In many applications there might be multiple agents or persons searching for solutions in the same solution space.
- Such scenarios usually occur in game playing where two opponents also called adversaries are searching for a goal.
- Their goals are usually contrary to each other.

Types of Games

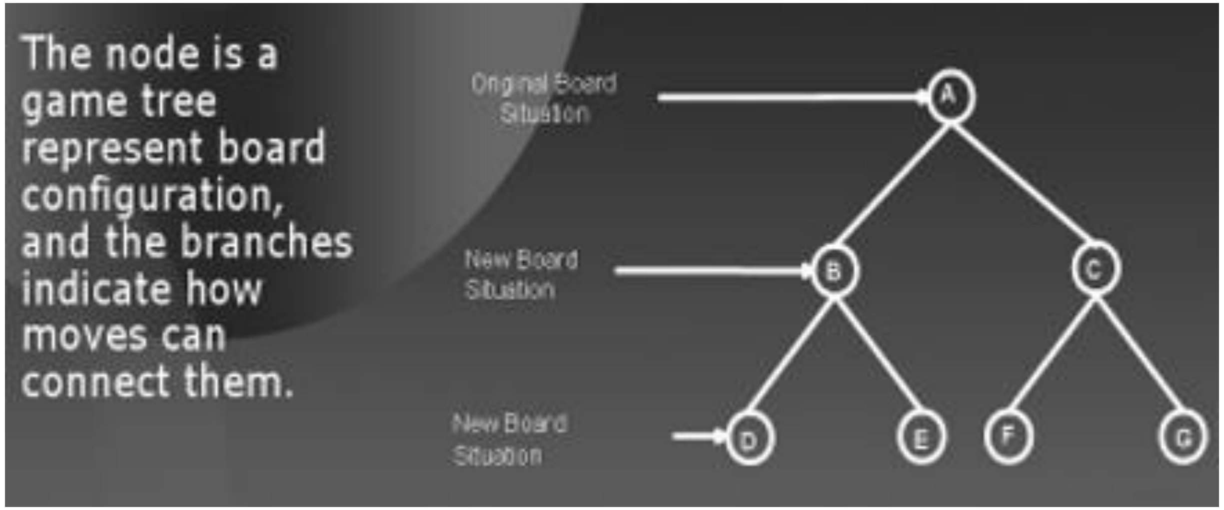
- We want to look at games that are
  - Deterministic
  - Finite
  - Discrete
  - Turn-taking
  - Two-player
  - Zero-sum
  - Perfect information

Types of Games

- Zero-sum Game – Gains of one player = Losses of the other player
- Discrete – Game states and decisions are finite
- Deterministic – No chance is involved e.g. no throwing of die
- Games with perfect information – Both players can see the states…each decision made sequentially…no simultaneous moves – Scrabble does not have perfect information available to the other player – Example: Chess, Go, Tic Tac Toe
- Finite – Finite number of states and decisions, e.g., tennis is not finite

Adversarial Search

- Many games can be modelled as trees as shown below. We will focus on board games for simplicity.
- Searches in which two or more players with contrary goals are trying to explore the same solution space in search of the solution are called adversarial searches.



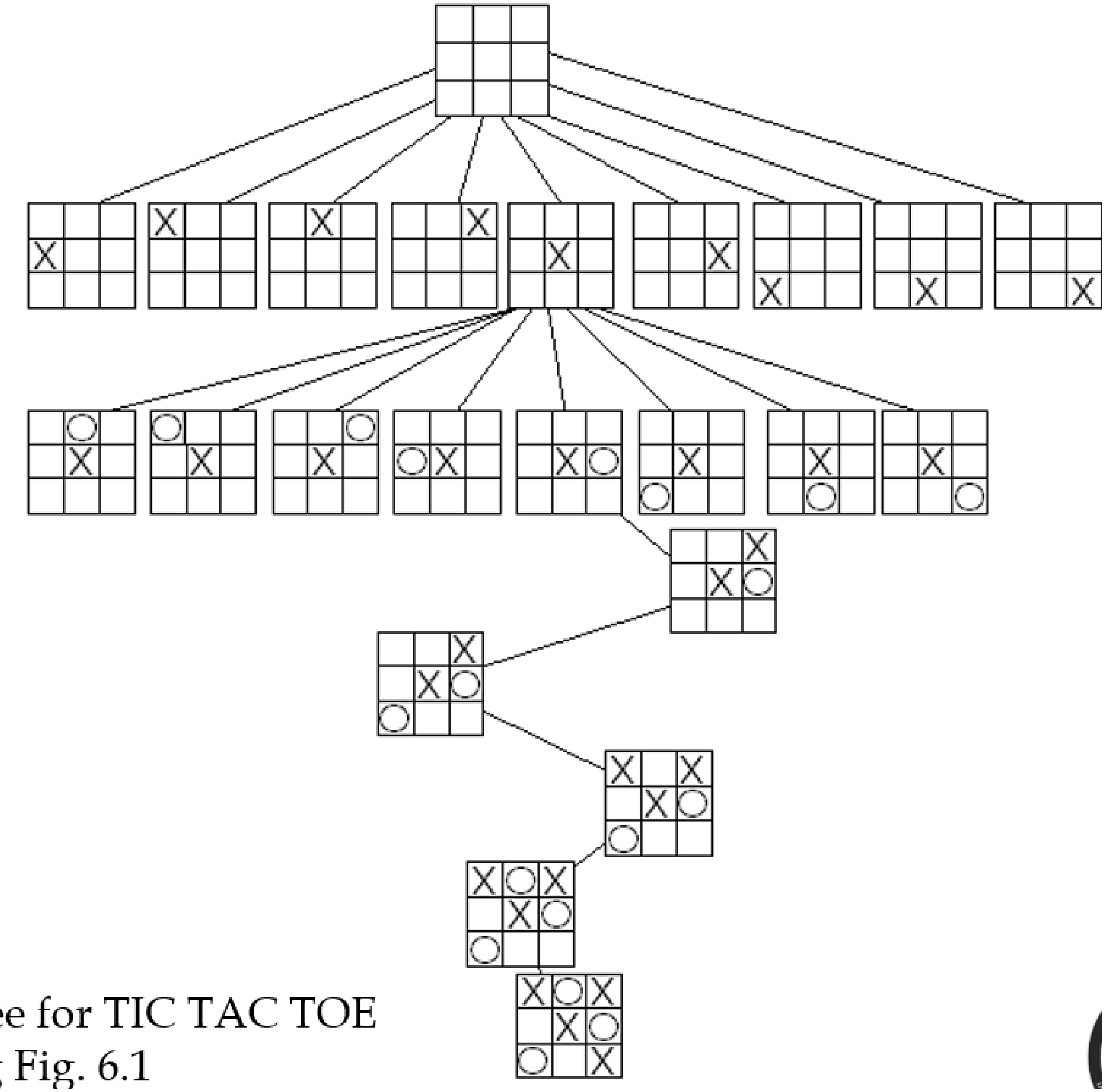
Game Trees

- Game can be defined as a kind of search problem
- Initial State S: Includes board position and identifies the player to move
- Successor Function Succ: Returns a list of (move, state) pairs each indicating a legal move and resulting state
- Terminal Test: Test that determines when a game is over.
- Terminal State T: State where the game has ended
- Utility Function/ Payoff Function/ Objective Function V: Function that gives a numeric value for the terminal state, e.g. Chess +1, -1, or 0 (win, loss, or draw)
- Game Tree: Initial state and legal moves for each side define a game tree for the game

Game Trees

- Nodes denote board configurations
- Branches indicate how one configuration can be transformed into another by a single legal move.
- A situation analyser that converts all judgments about board situations into a single, over all quality number.
- This situation analyser is also called a static evaluator and the score/ number calculated by the evaluator is called the static evaluation of that node.
- Positive numbers, by convention indicate favour to one player.
- Negative numbers indicate favour to the other player.

Game Trees



Part of the search tree for TIC TAC TOE
See Russel & Norvig Fig. 6.1

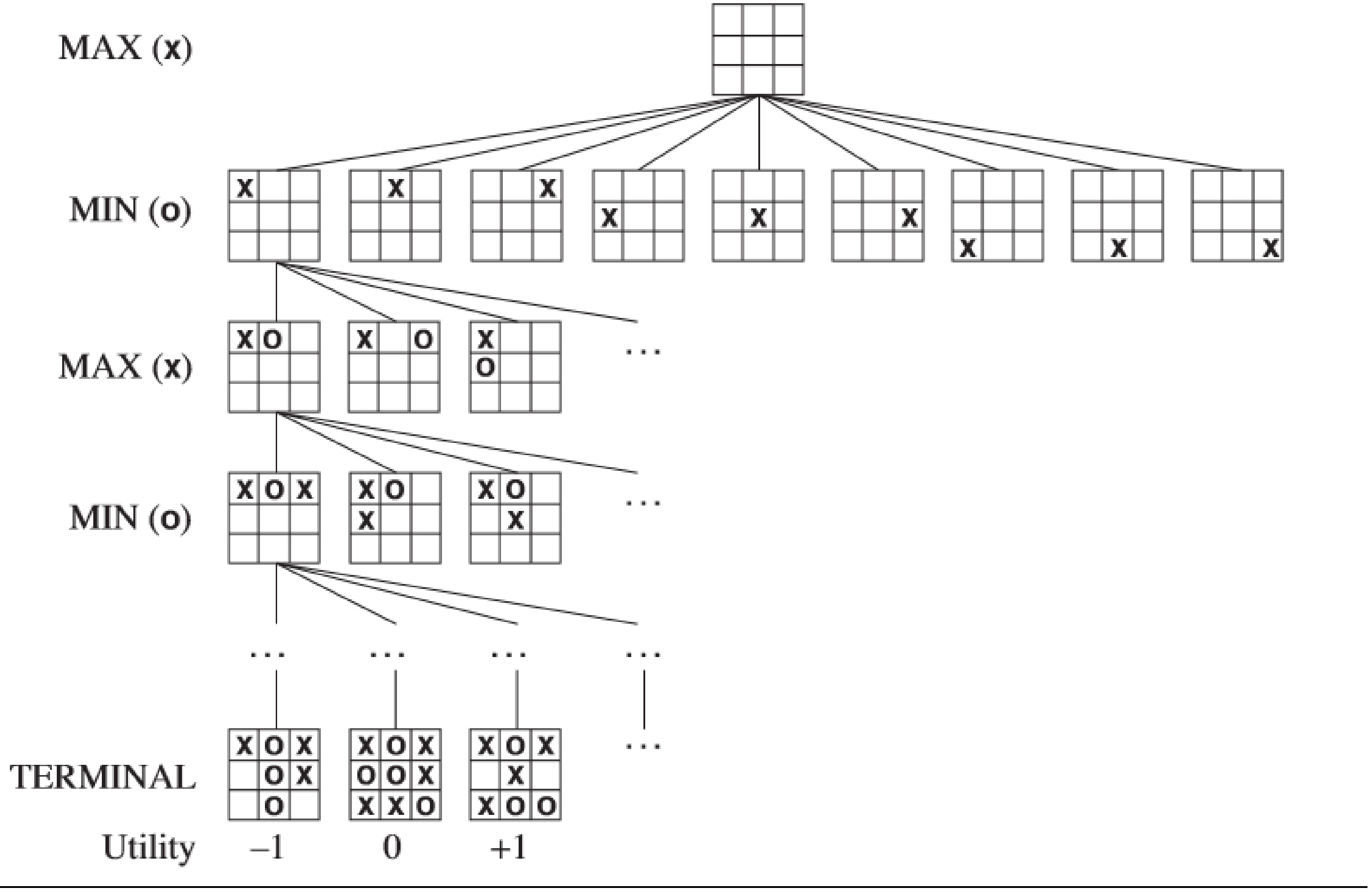
Minimax Procedure

- Opponents in a game are MAX and MIN
- MAX or maximizer is the player trying to win and maximize his/her advantange.
- MIN or minimizer is the opponent attempting to minimize max’s score
- We have a zero sum game
- MIN uses the same information and tries to move to a state that is worst for MAX

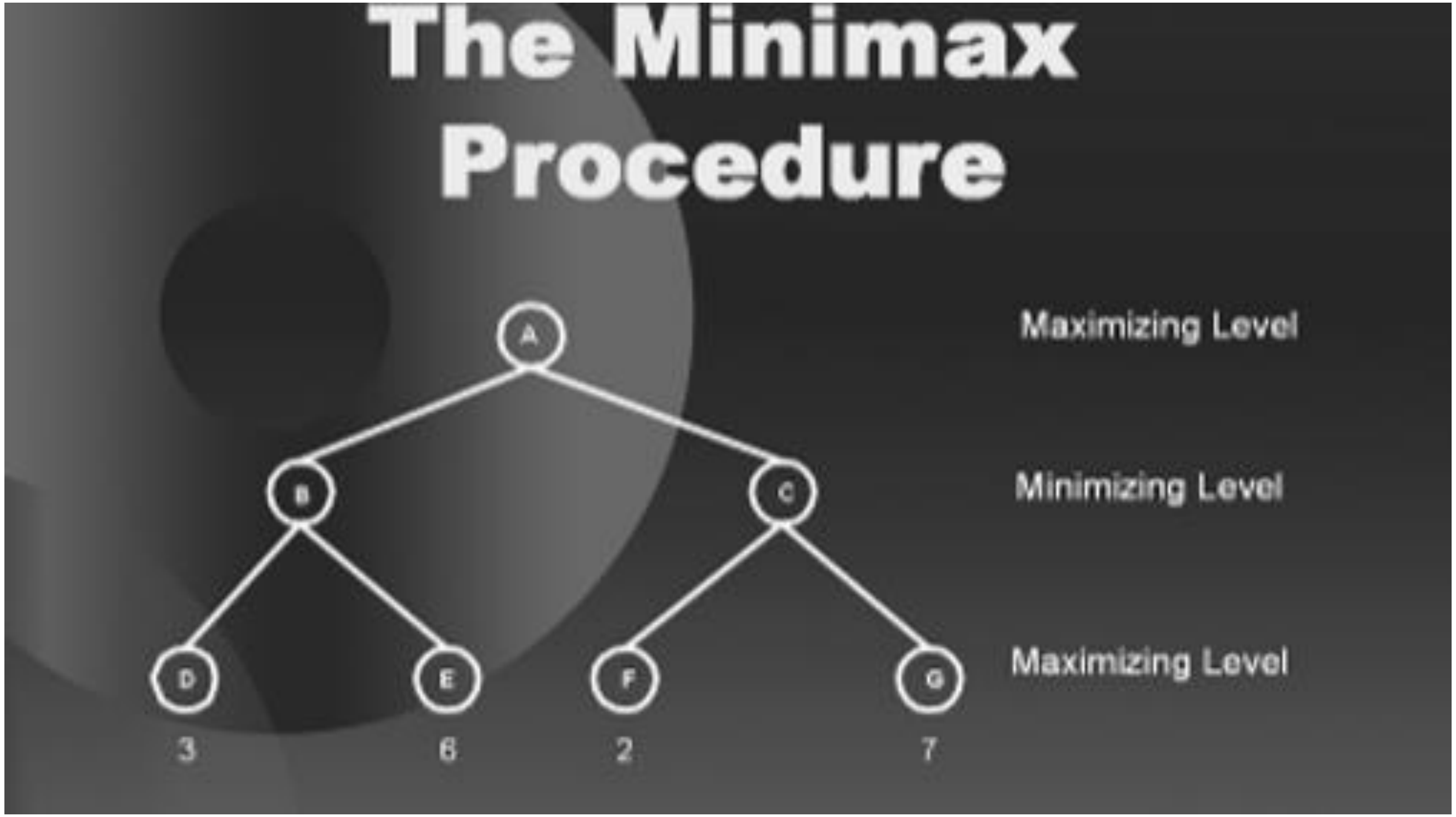
Minimax Procedure

- **ALGORITHM**
- When limit of search is reached, compute static value of current position relative to appropriate player.
- If level is minimizing level use minimax on children of current position and report minimum of result.
- If level is maximizing level use minimax on children of current position and report maximum of result.

Minimax Procedure



Minimax Procedure



Minimax Procedure

- Standing at node A the maximizer wants to decide which node to visit next, that is, choose between B or C.
- The maximizer wishes to maximize the score so apparently 7 being the maximum score, the maximizer should go to C and then to G.
- But when the maximizer will reach C the next turn to select the node will be of the minimizer, which will force the game to reach configuration/node F with a score of 2.
- Hence maximizer will end up with a score of 2 if he goes to C from A.
- On the other hand, if the maximizer goes to B from A the worst which the minimizer can do is that he will force the maximizer to a score of 3.
- Now, since the choice is between scores of 3 or 2, the maximizer will go to node B from A.

Minimax Procedure

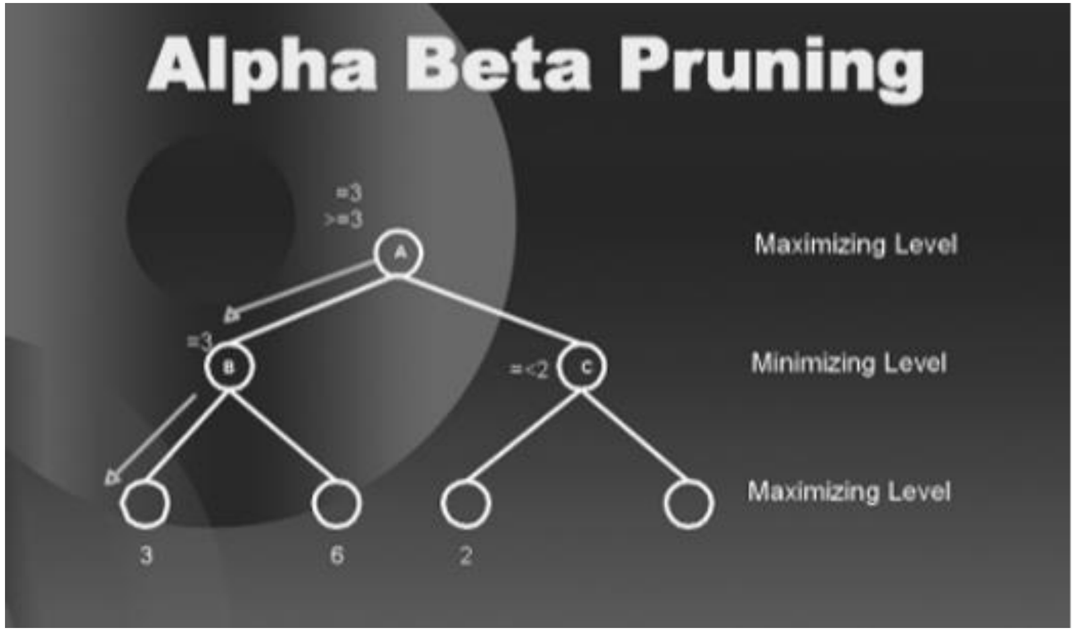
- The procedure by which scoring information passes up the game tree is called MINIMAX procedure, because the scores are either minimum or maximum of the scores of the nodes below.
- Minimax procedure can be quite expensive.
- Generation of paths or static evaluation can require a lot of computation.
- The problem with minimax search is that the number of game states it has to examine is exponential in the depth of the tree.
- Chess has a branching factor of 35, with 35^100 nodes = 10^154

Alpha Beta Pruning

- In Minimax Procedure, it seems as if the static evaluator must be used on each leaf node.
- Fortunately there is a procedure that reduces both the tree branches that must be generated and the number of evaluations.
- This procedure is called Alpha Beta pruning which “prunes” the tree branches thus reducing the number of static evaluations.
- We use the following example to explain the notion of Alpha Beta Pruning.
- Suppose we start of with a game tree in the diagram below.
- Notice that all nodes/situations have not yet been previously evaluated for their static evaluation score.
- Only two leaf nodes have been evaluated so far.

Alpha Beta Pruning

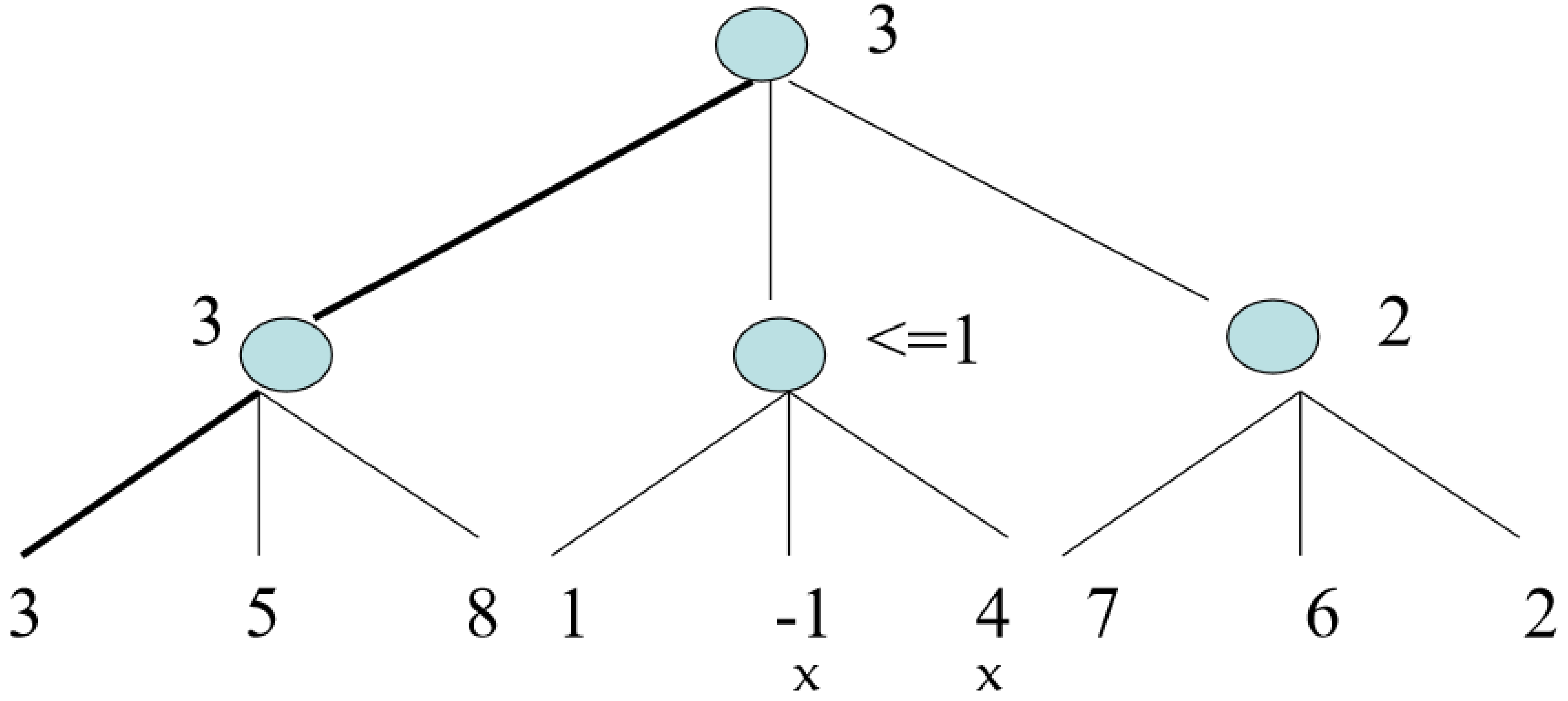
- Notice that all nodes/situations have not yet been previously evaluated for their static evaluation score.
- Only two leaf nodes have been evaluated so far.



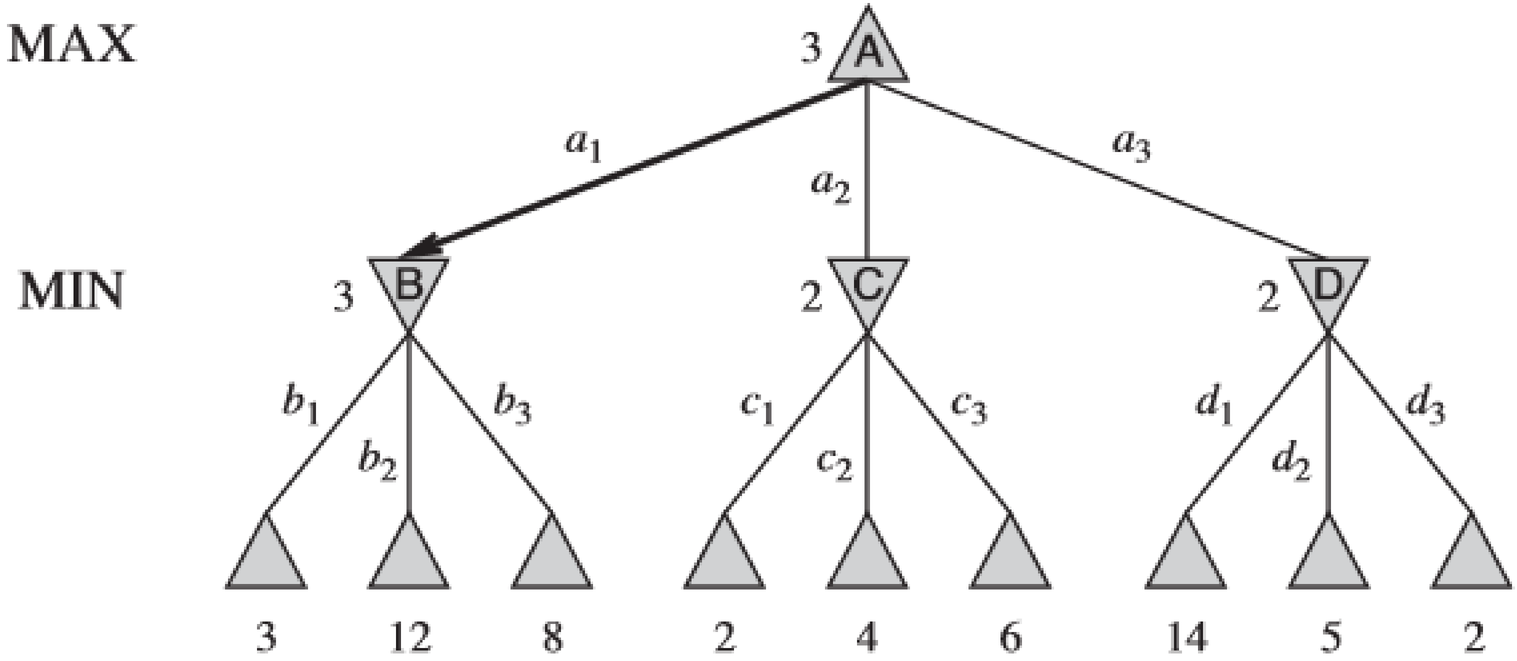
Alpha Beta Pruning

- Sitting at A, the player-one will observe that if he moves to B the best he can get is 3.
- So the value three travels to the root A.
- Now after observing the other side of the tree, this score will either increase or will remain the same as this level is for the maximizer.
- When he evaluates the first leaf node on the other side of the tree, he will see that the minimizer can force him to a score of less than 3 hence there is no need to fully explore the tree from that side.
- Hence the right most branch of the tree will be pruned and won’t be evaluated for static evaluation.

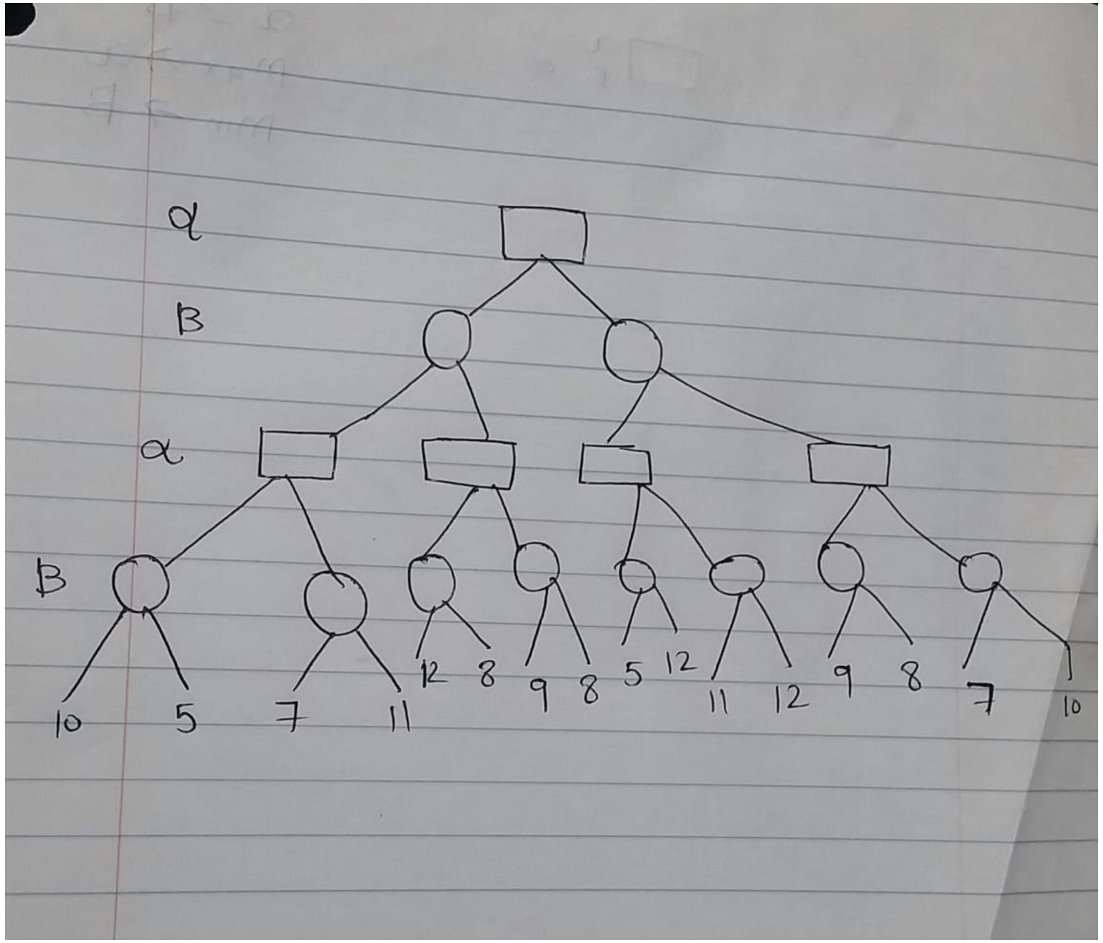
Alpha Beta Pruning



Alpha Beta Pruning



Alpha Beta Pruning



Alpha Beta Pruning

- Reduces the number of tree branches that must be generated
- Reduces the number of static evaluations that must be made
- Order of branches is most important
- If worst paths examined first then no cutoffs occur.
- What is the BEST case for alpha-beta pruning???
- What is the WORST case for alpha-beta pruning???

End of Slides