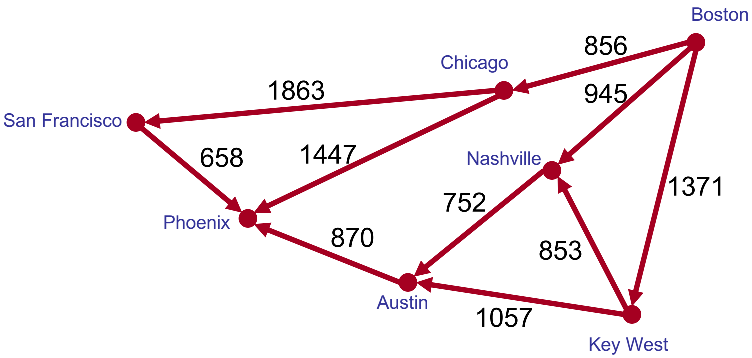
Adversarial Search

CPSC 470 – Artificial Intelligence

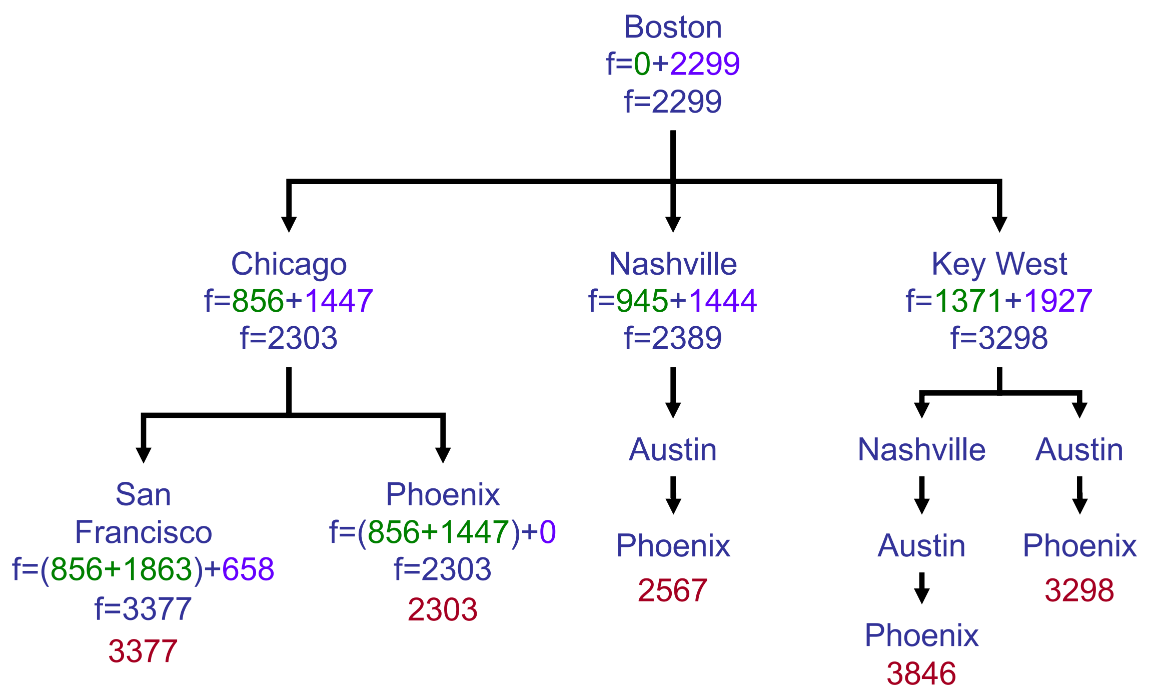
Brian Scassellati

A* Search



- Combine Greedy search with Uniform Cost Search
- Minimize the total path cost (f) =
  actual path so far (g) +
  estimate of future path to goal (h)

	Distance to Phoenix
Boston	2299
Chicago	1447
Nashville	1444
Key West	1927
Austin	870
San Francisco	658

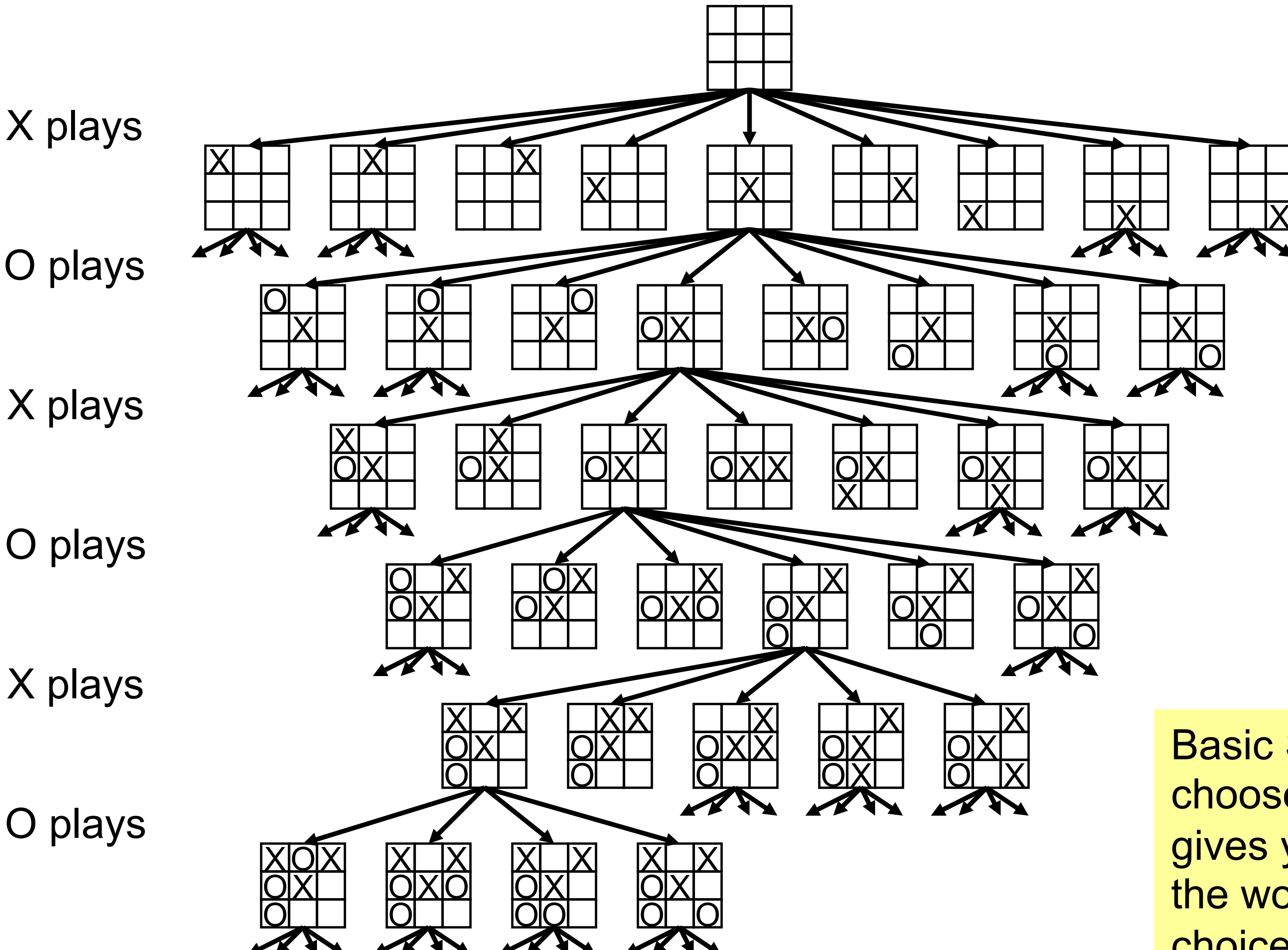


Total Distance Flown

What if you can't control the path taken through the search tree?

(How to play games and make it look like research…)

A Partial Search Tree for Tic-Tac-Toe

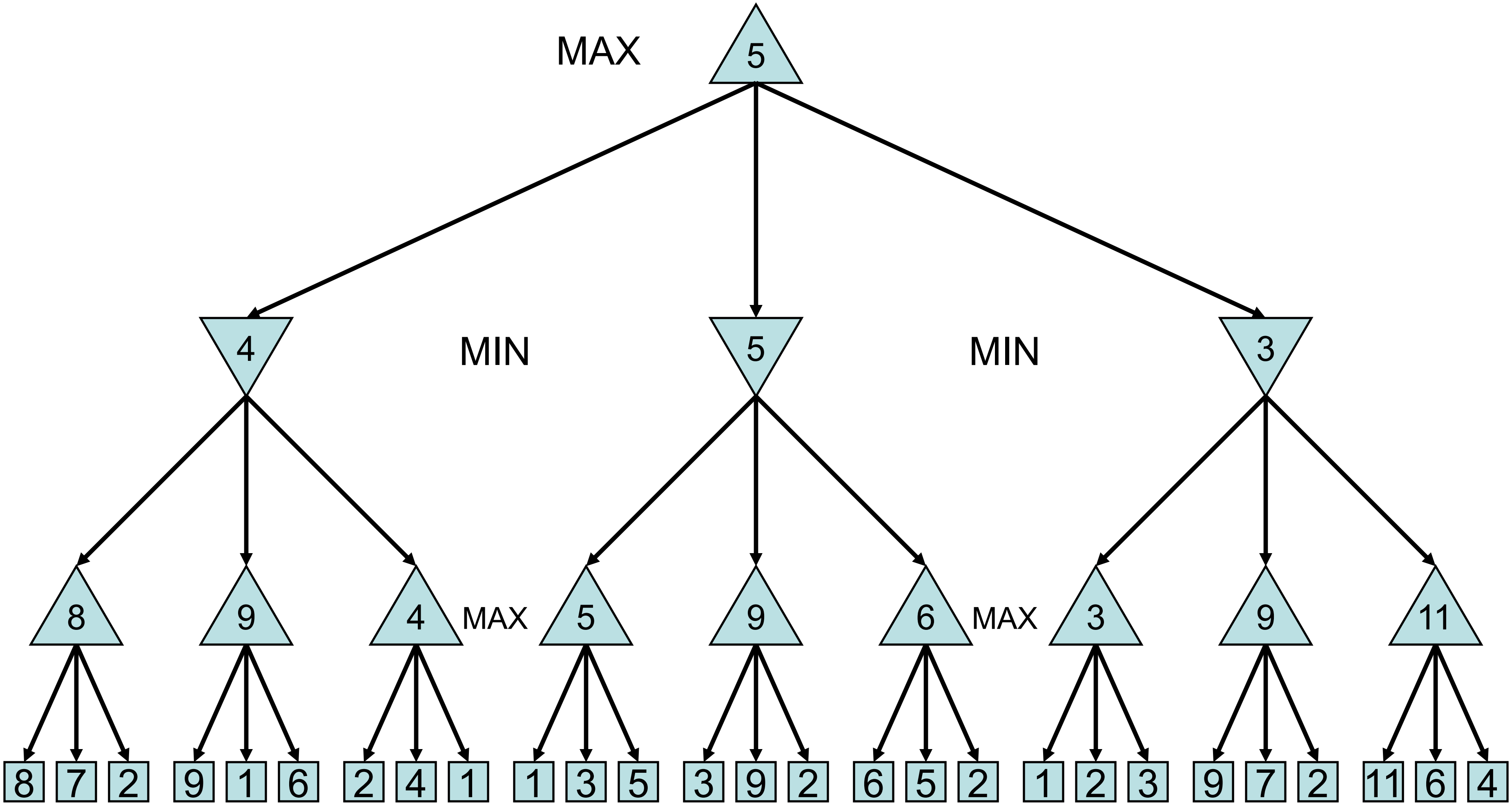


Basic Strategy: choose a move that gives your opponent the worst set of choices

The Minimax Algorithm

1. Generate the entire game tree
2. Apply the utility function to each terminal node (high values are good for your side)
3. Filter values from the terminal nodes up through the tree:
   a. At nodes controlled by your opponent, choose the minimum value of the children
   b. At nodes controlled by you, choose the maximum value of the children
4. When you reach the top of the tree, you have an optimal solution

Minimax Example



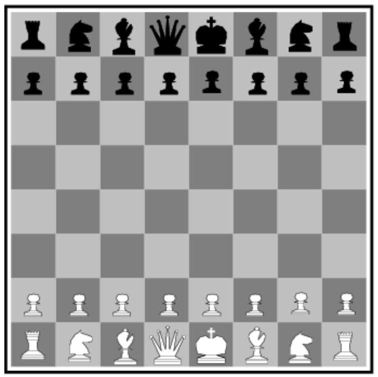
Is it practical to construct a complete search tree?

- Typical chess program using minimax
  - Evaluate 1000 positions per second
  - Tournament chess is 150 seconds per move
  - Total of 150,000 positions
  - Branching factor for chess is ~35
  - Evaluate only 3-4 ply
  - Average human player can make plans 6-8 ply ahead

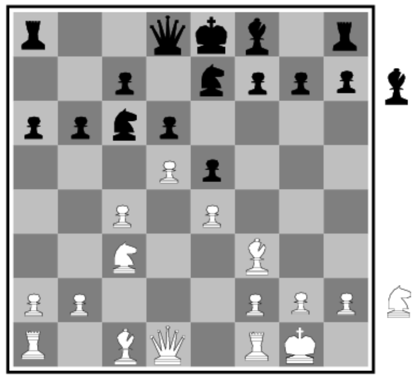
Imperfect Decisions

- What if you don’t have time/space to build the entire search tree?
  - Use a heuristic and limit the depth!
    - In game playing, the heuristic function is often called an evaluation function
  - As always, the quality of the heuristic function can make an enormous impact

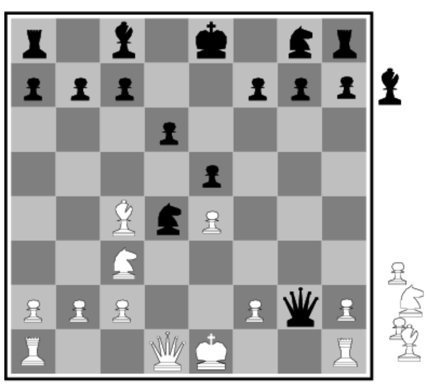
What do these Evaluation Functions Look Like?



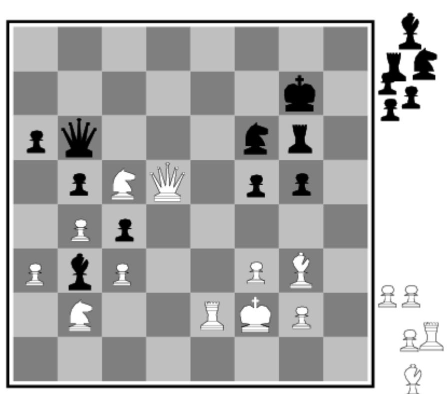
(a) White to move
Fairly even



(b) Black to move
White slightly better



(c) White to move
Black winning



(d) Black to move
White about to lose

- f(state)→real
- These heuristics are critical for complex games, like chess
- Account for
  - Piece count
  - Whose turn it is
  - Board positioning
- The relative ordering of values matters, not the values themselves

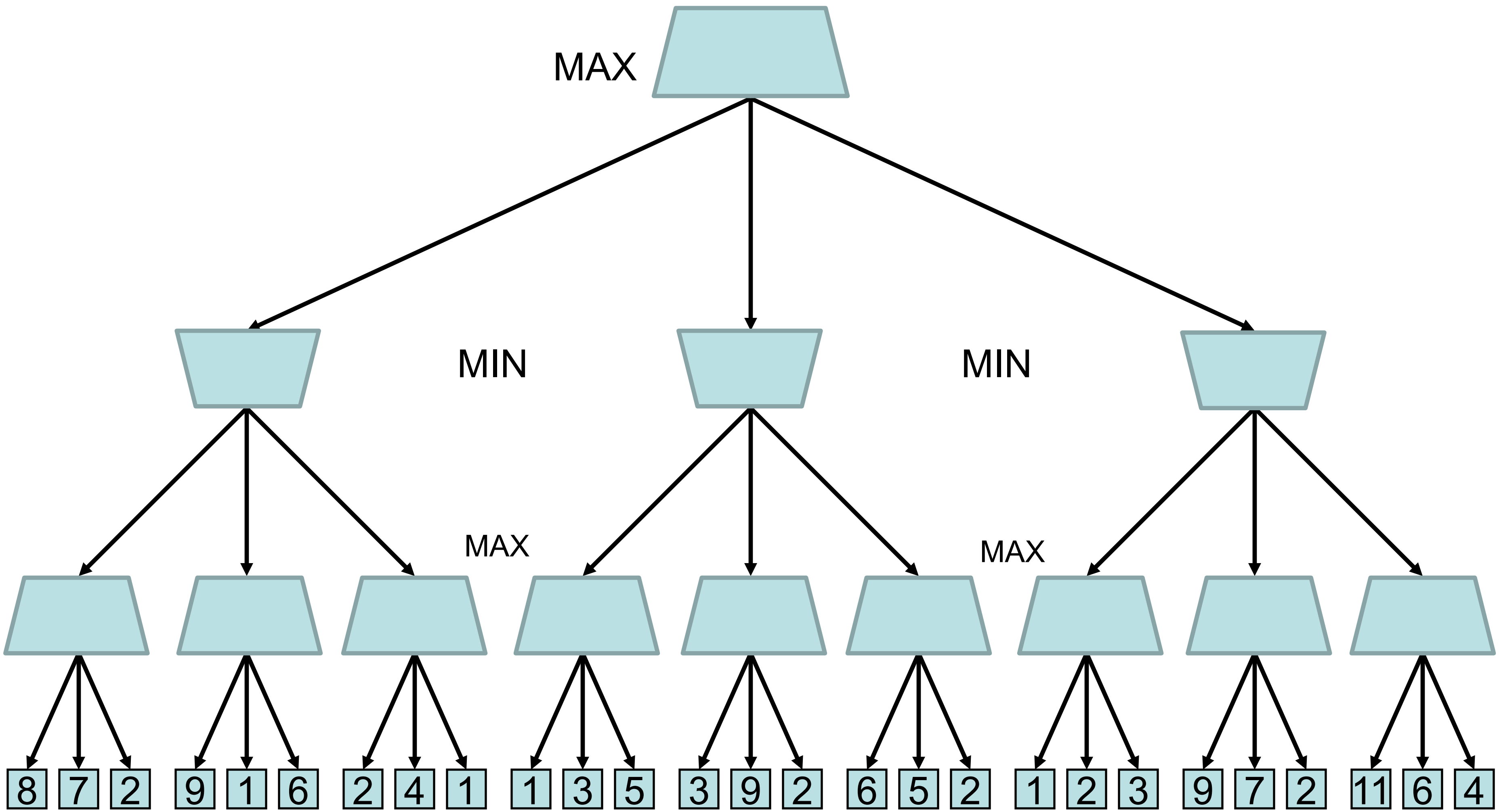
How to Improve Opponent Search: Pruning

- Don’t evaluate all the parts of the tree
- Pruning techniques eliminate parts of the search tree without looking at them
- Today, we will look at one simple but effective form of pruning: alpha-beta

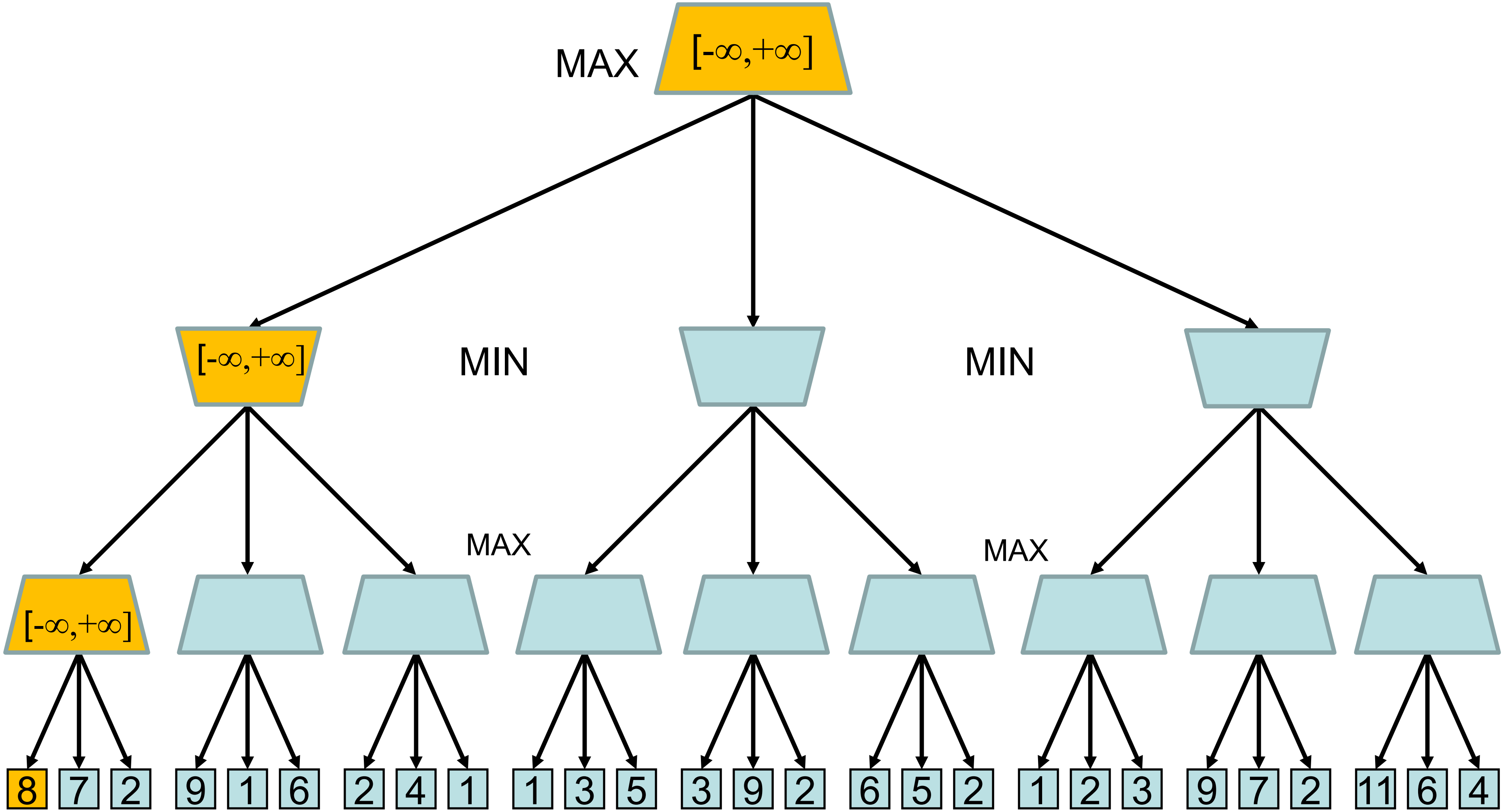
The alpha-beta principle

If you have an idea that is surely bad, don’t take time to see how truly awful it is

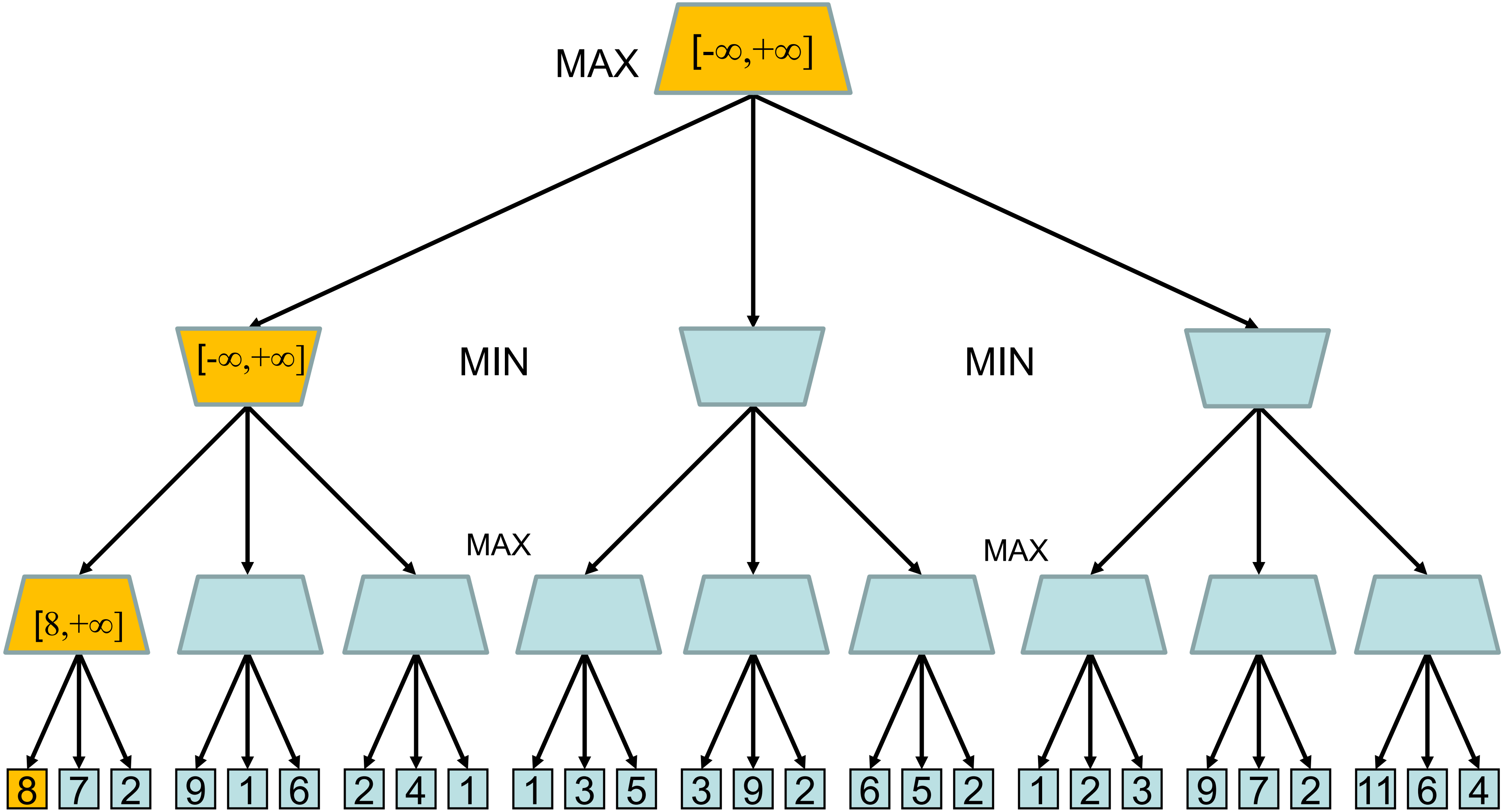
Alpha-Beta Pruning Example



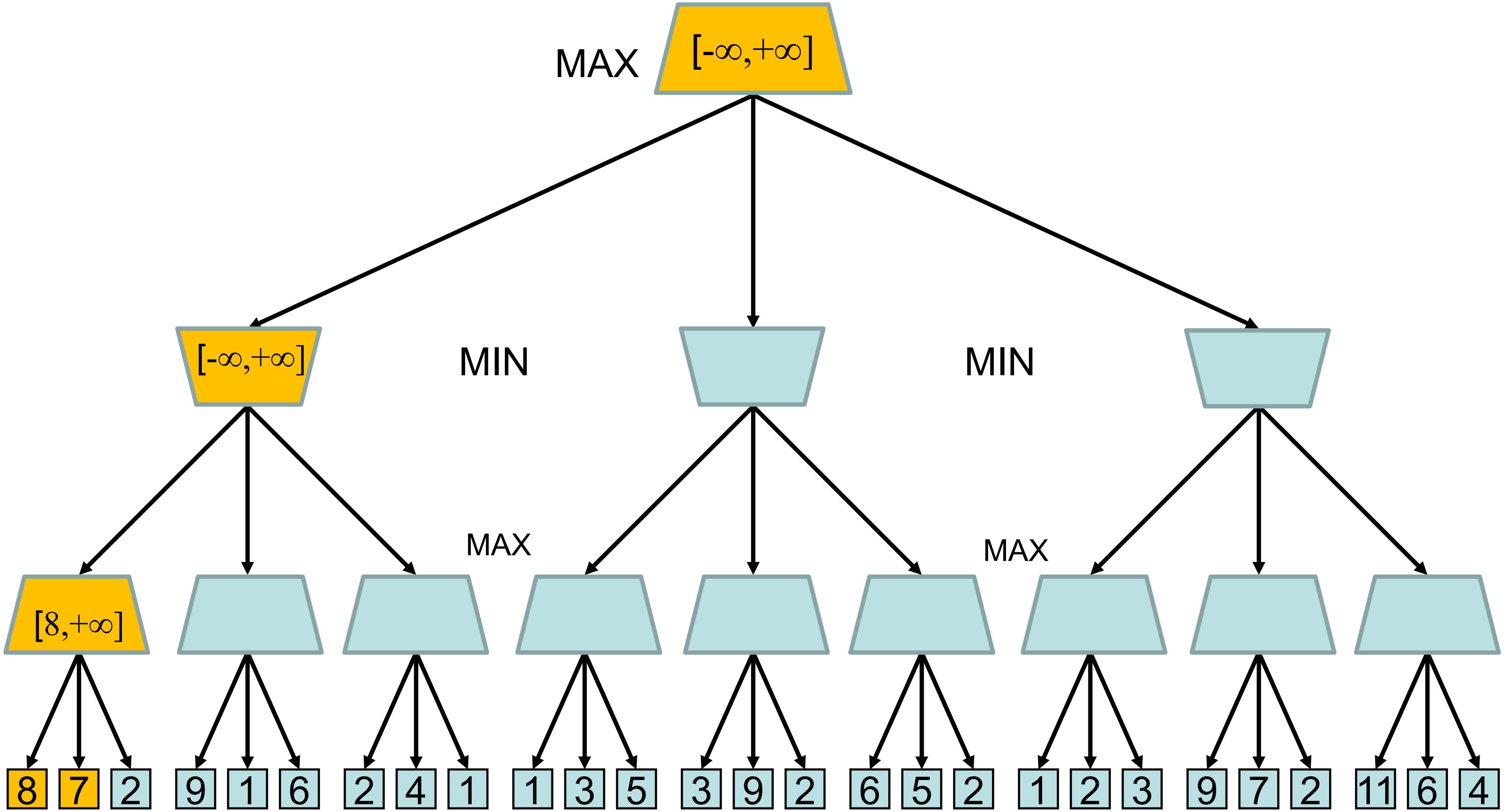
Alpha-Beta Pruning Example



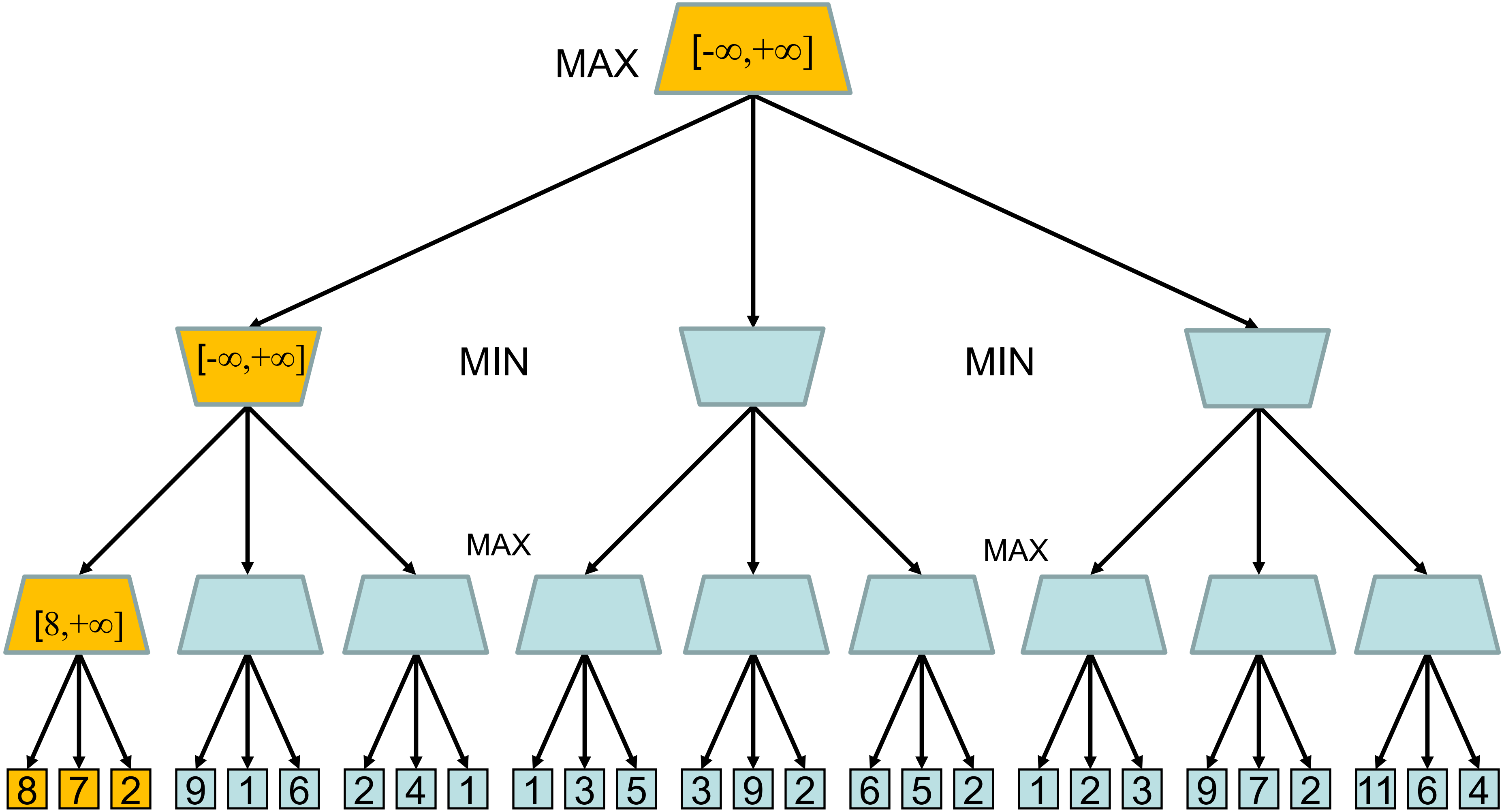
Alpha-Beta Pruning Example



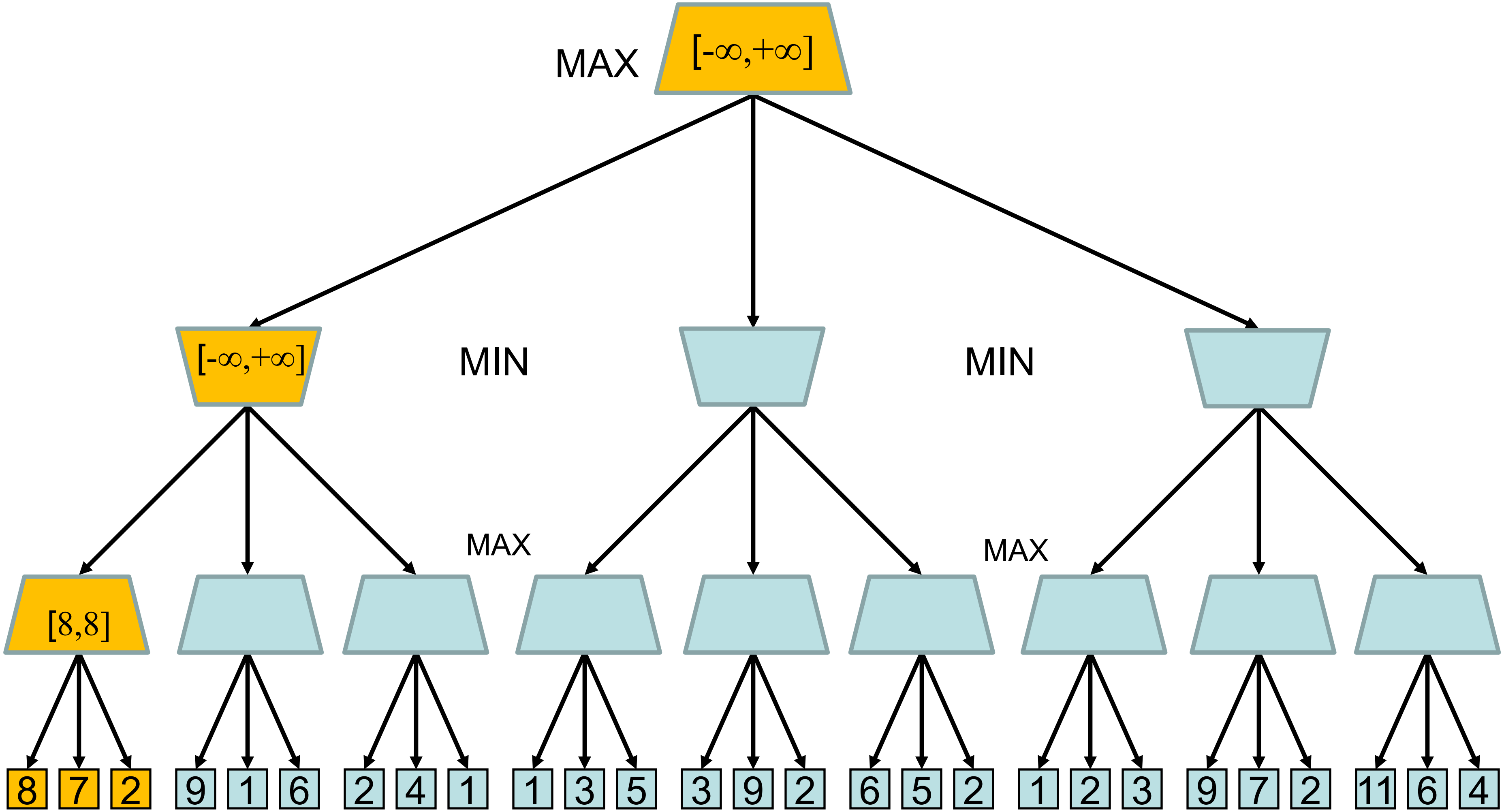
Alpha-Beta Pruning Example



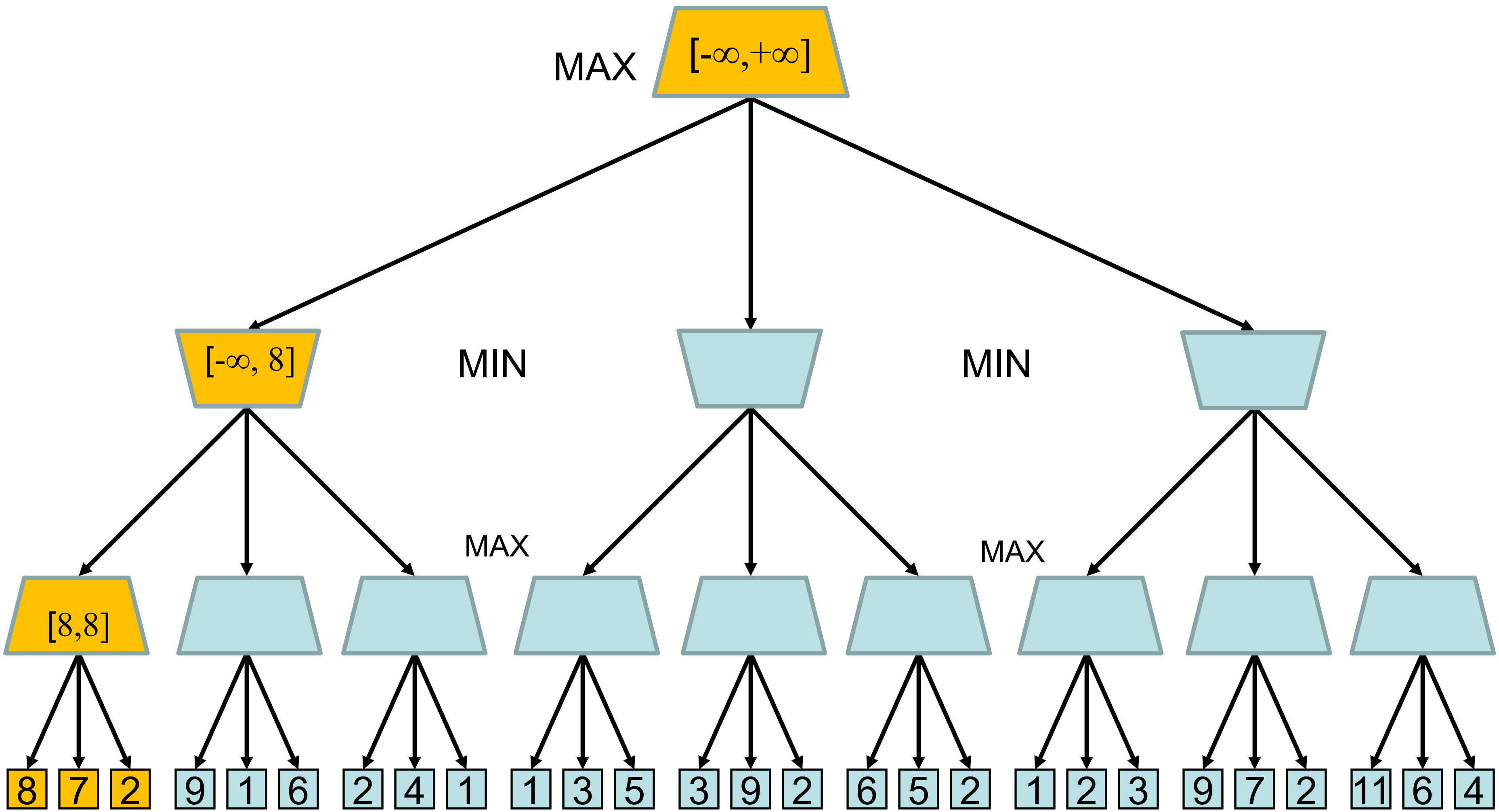
Alpha-Beta Pruning Example



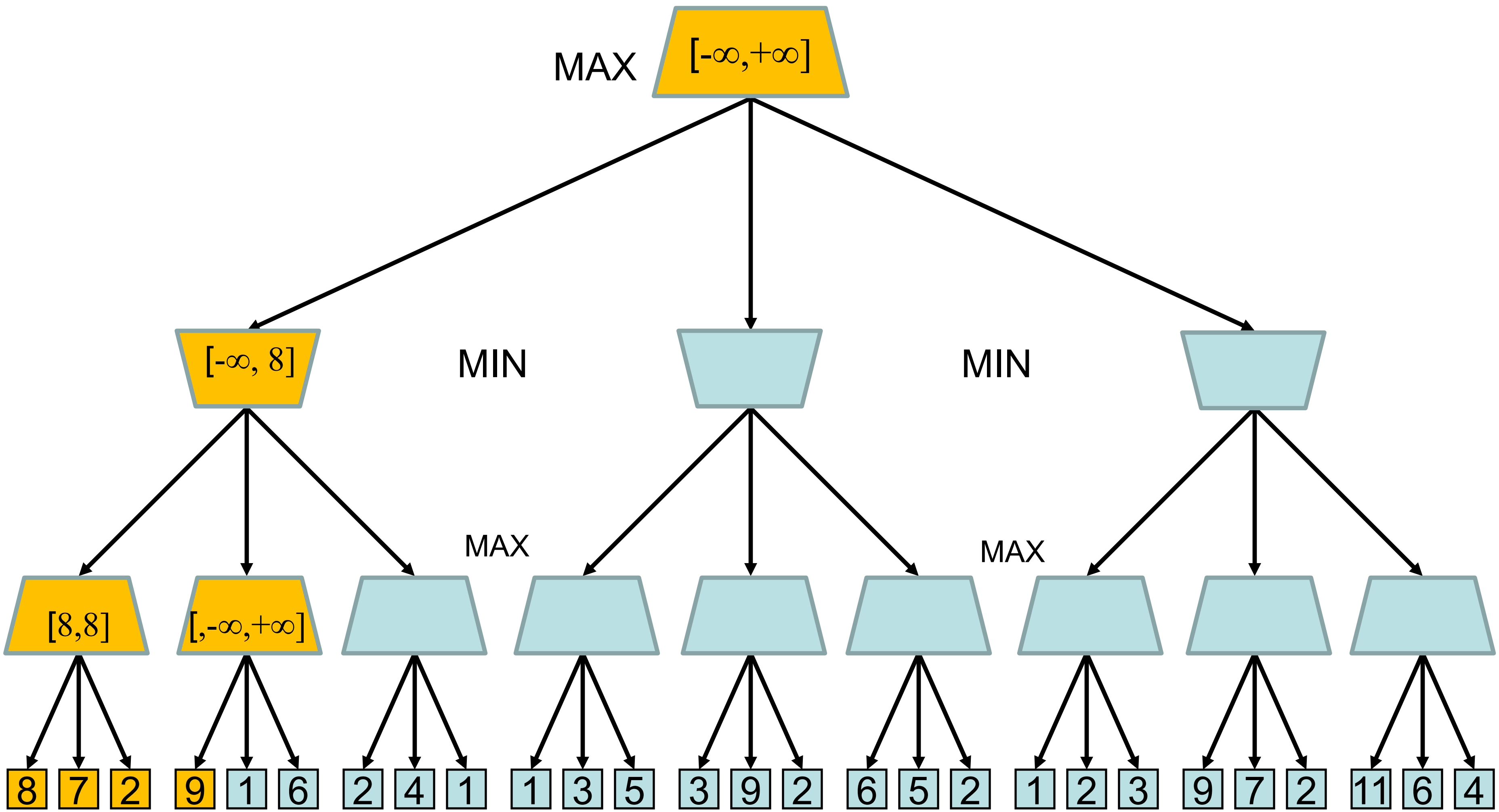
Alpha-Beta Pruning Example



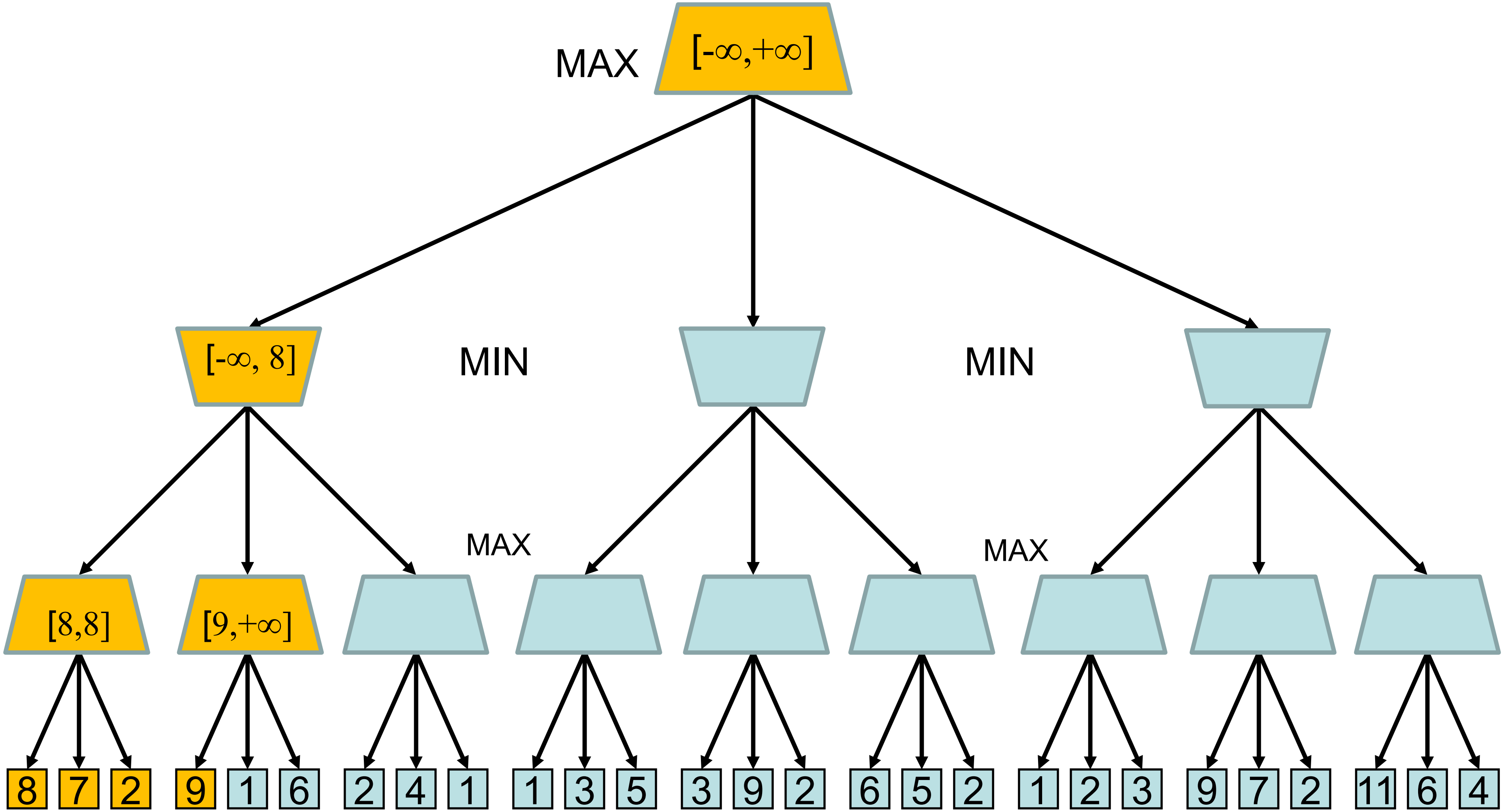
Alpha-Beta Pruning Example



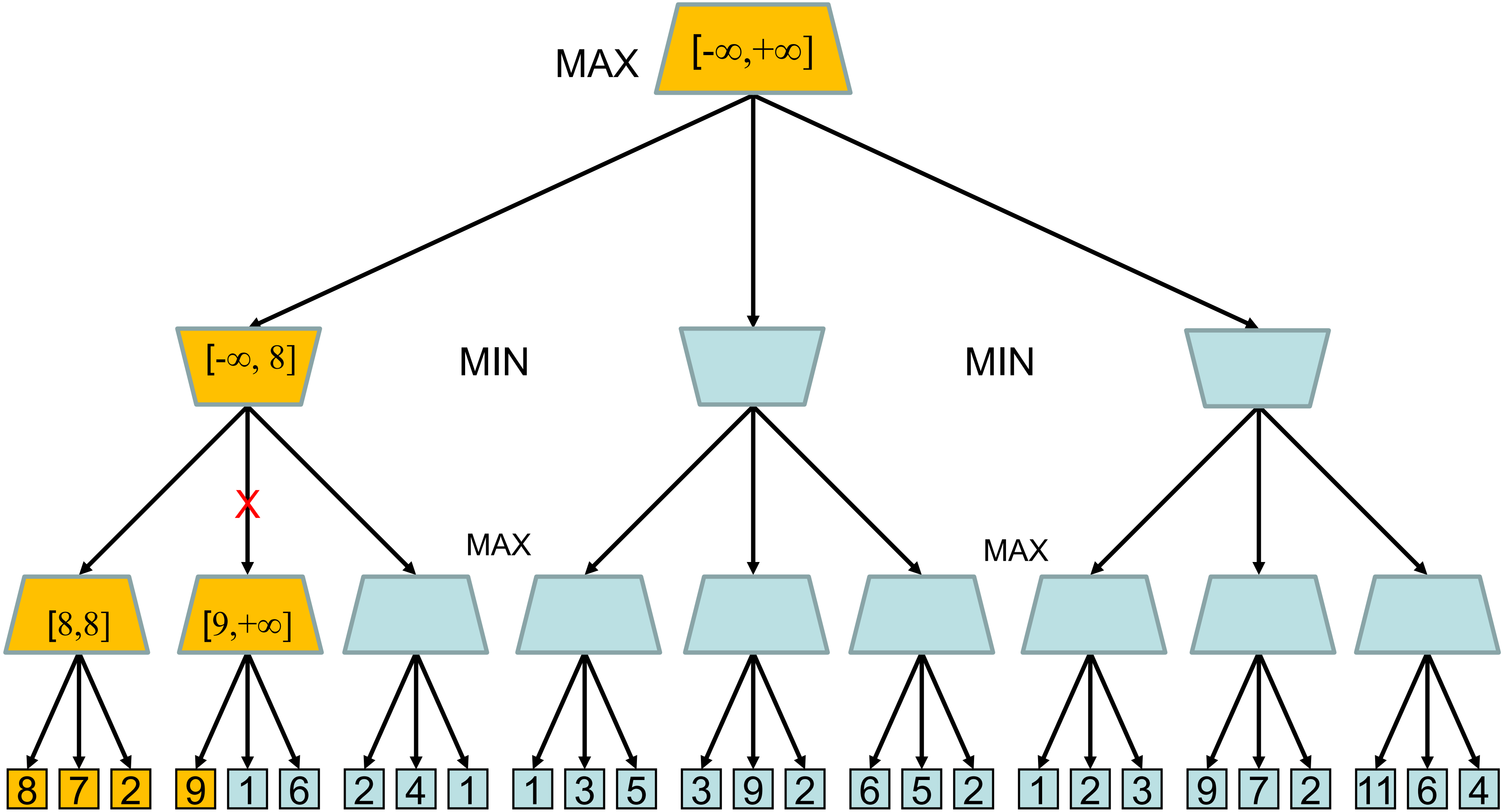
Alpha-Beta Pruning Example



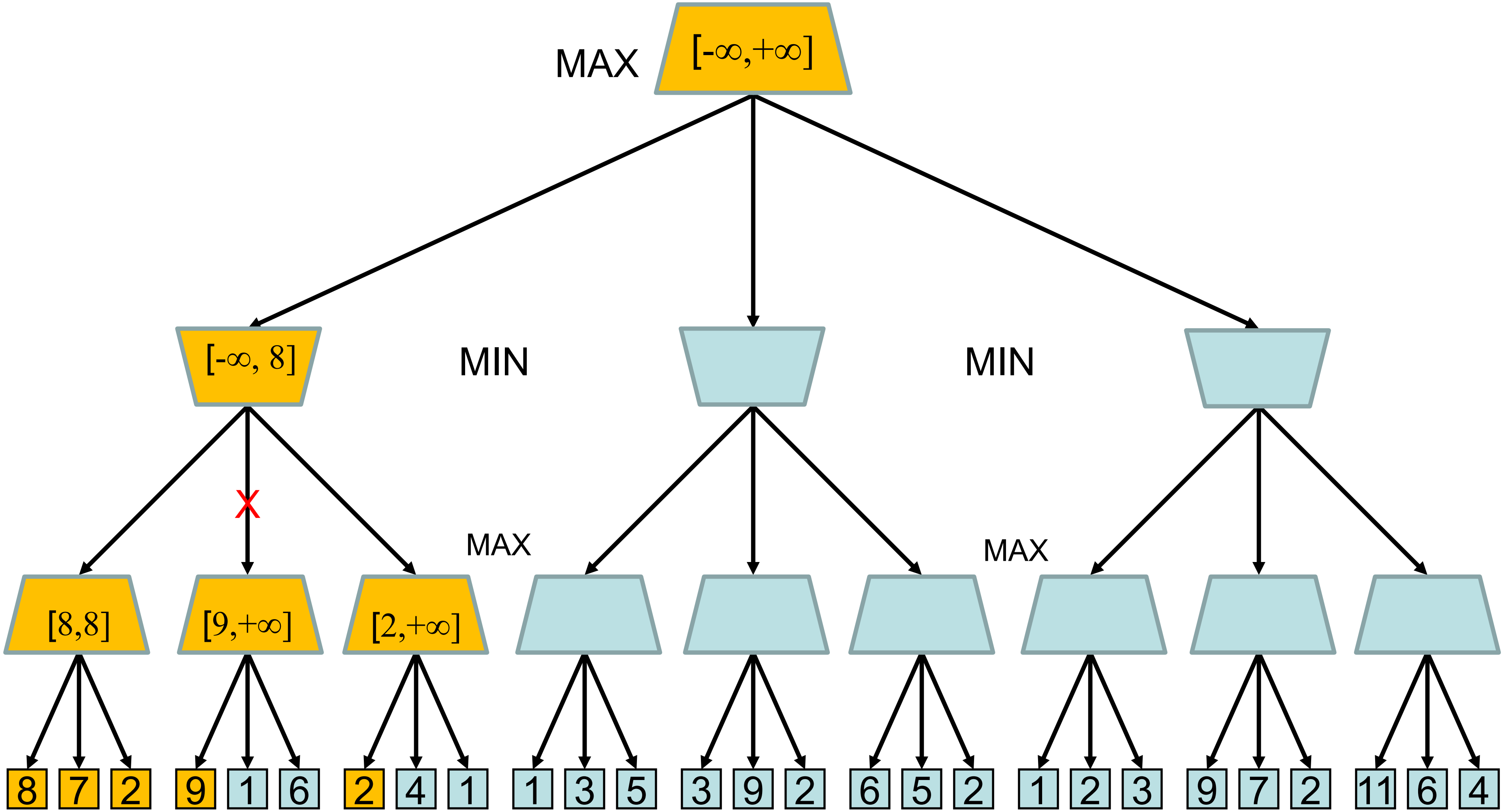
Alpha-Beta Pruning Example



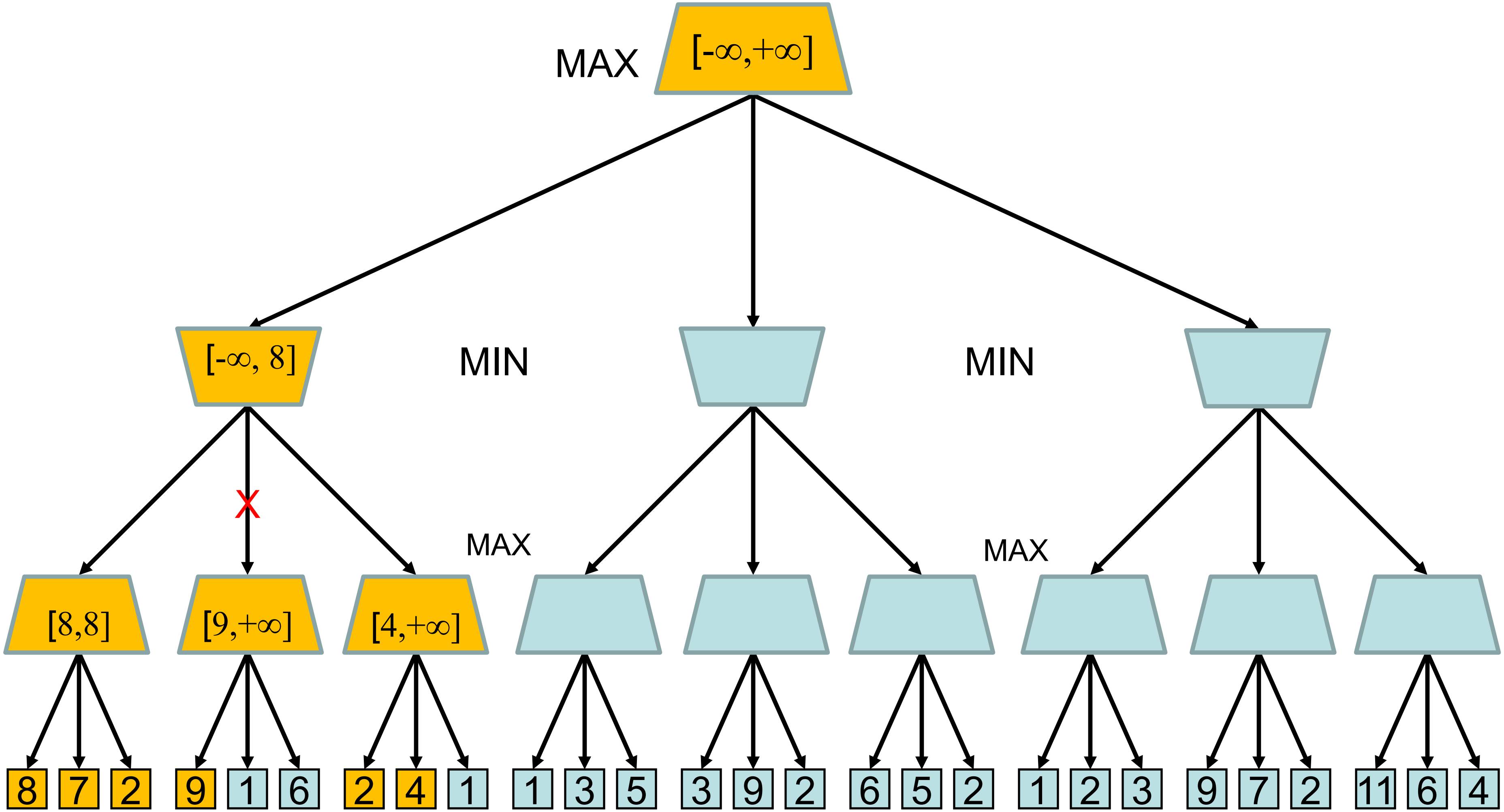
Alpha-Beta Pruning Example



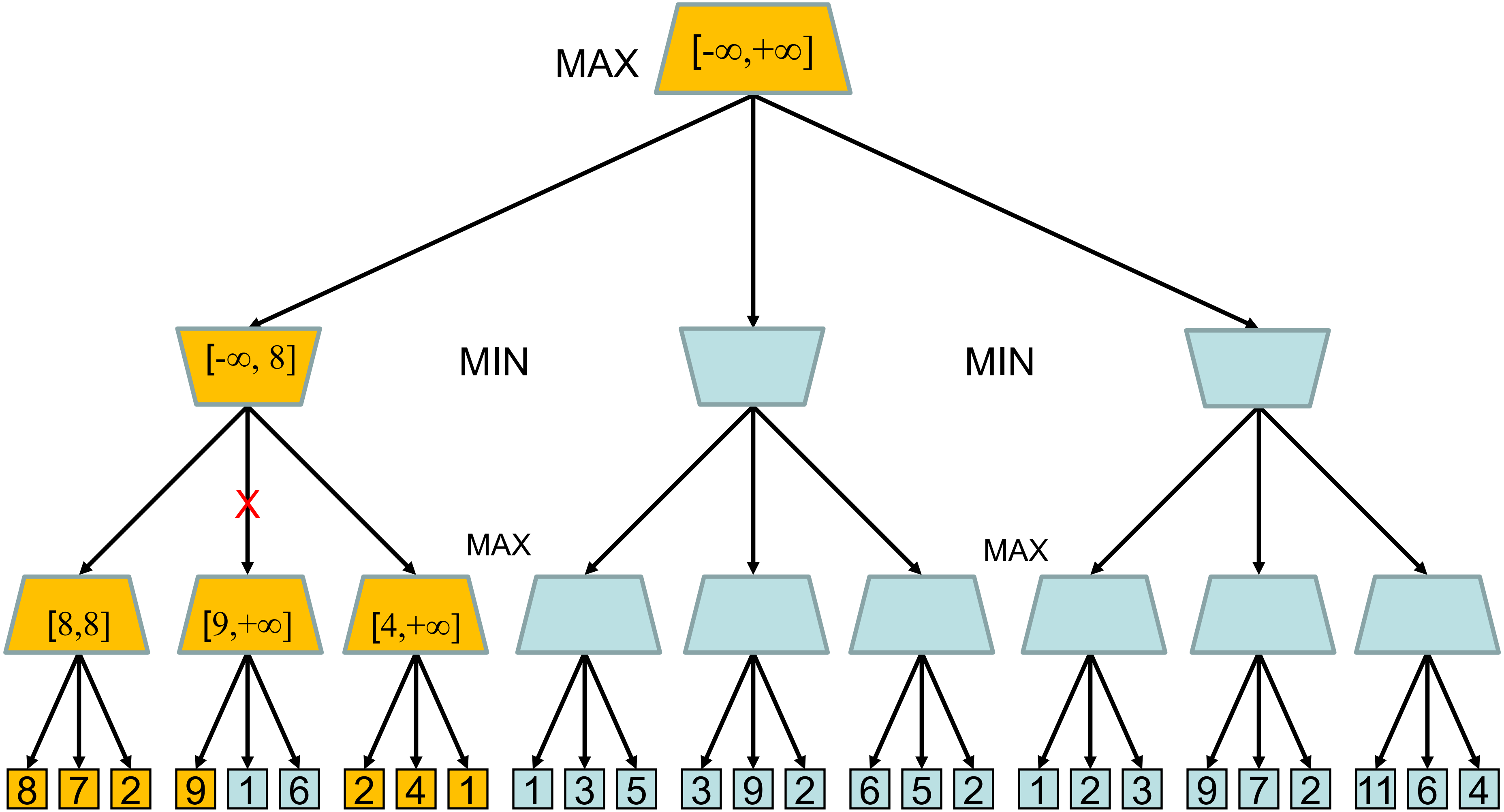
Alpha-Beta Pruning Example



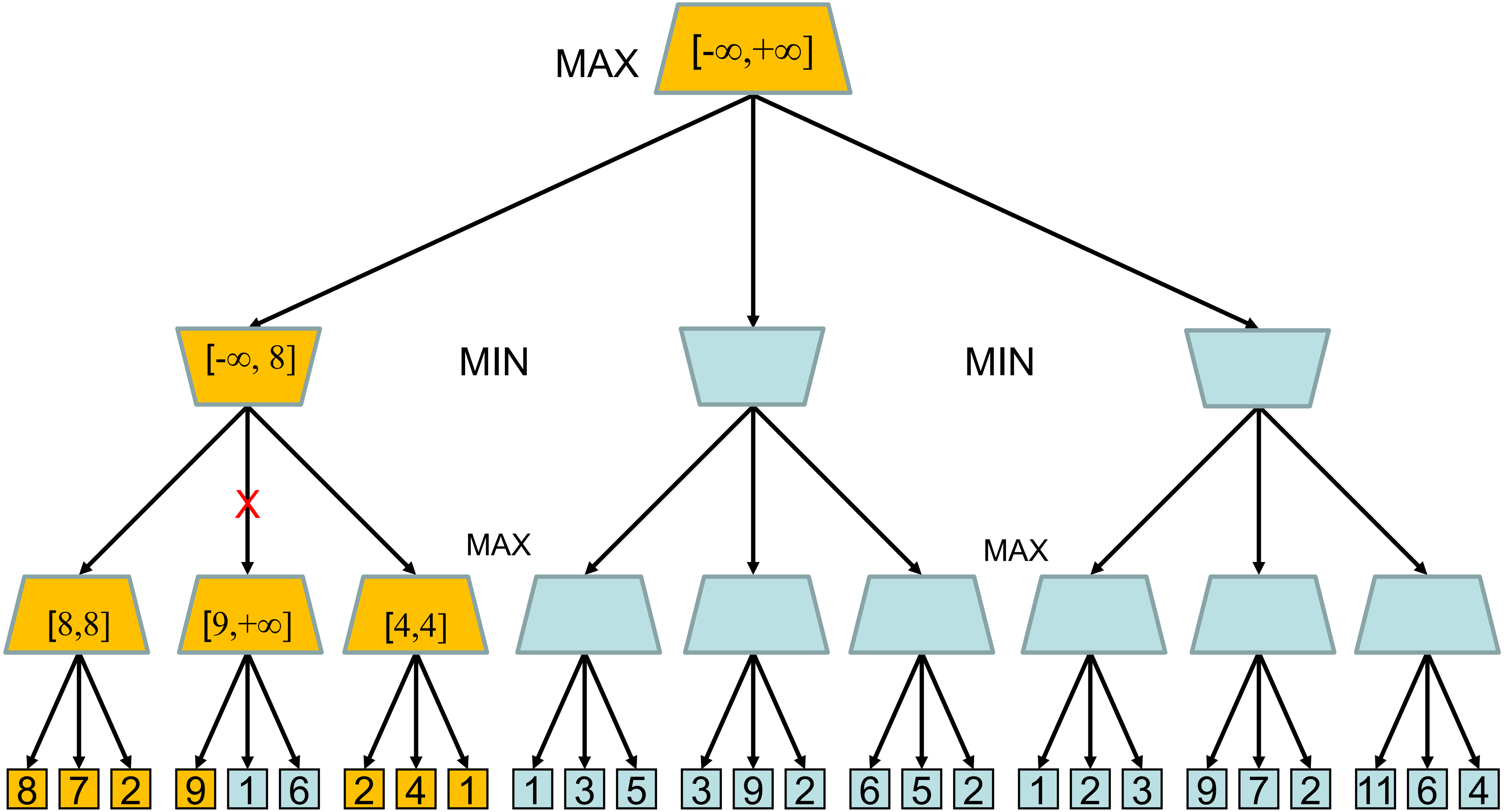
Alpha-Beta Pruning Example



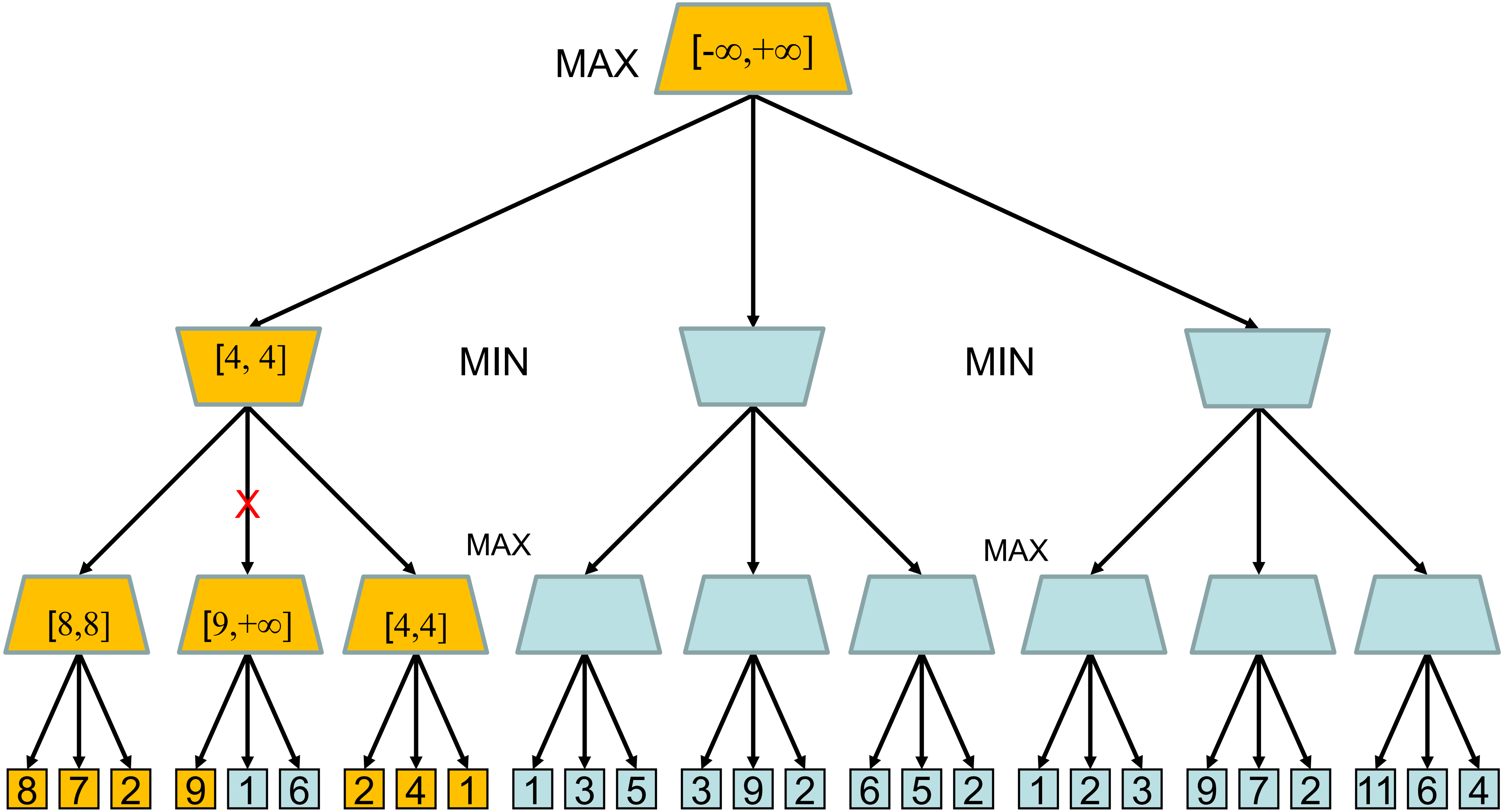
Alpha-Beta Pruning Example



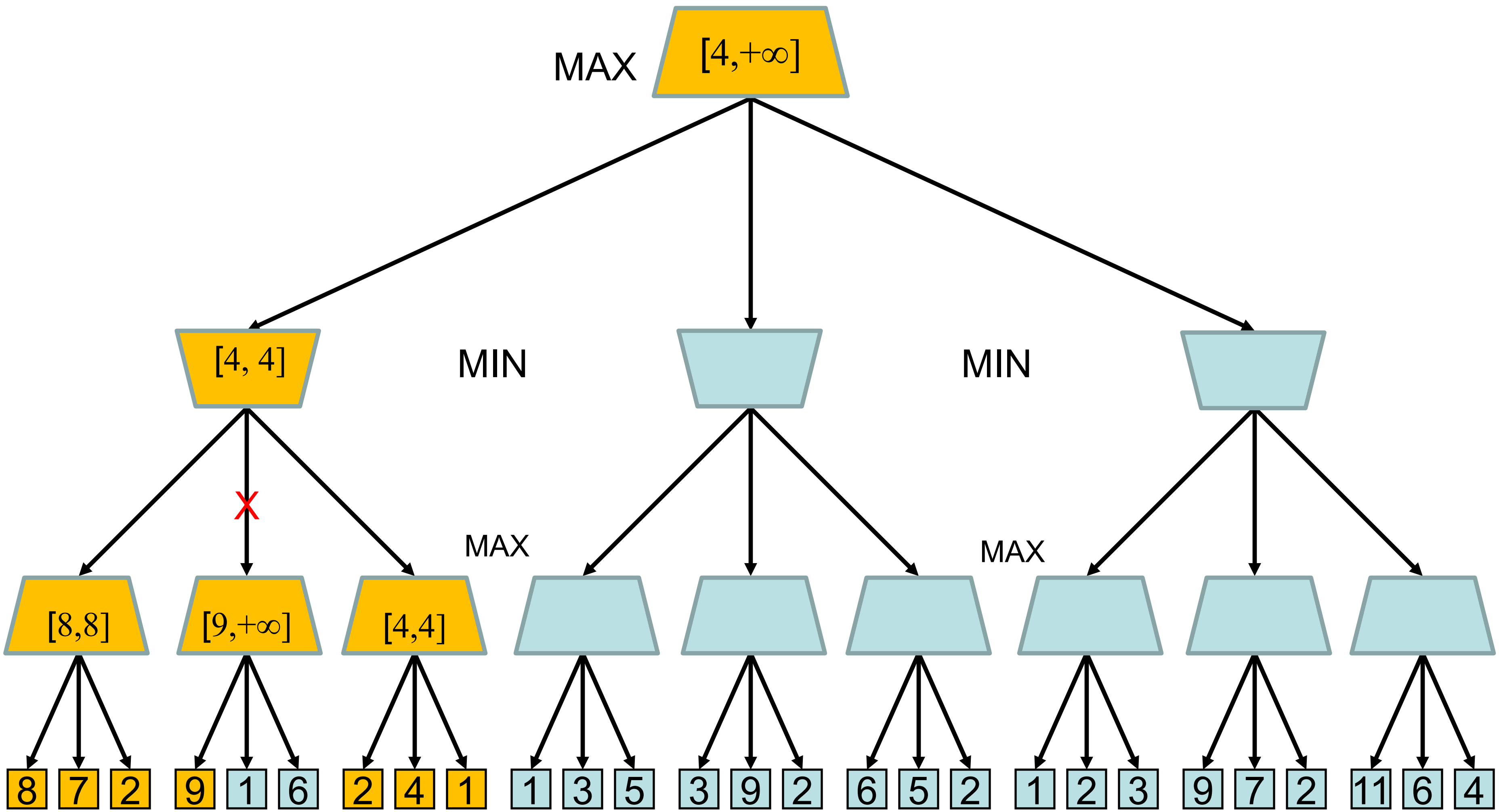
Alpha-Beta Pruning Example



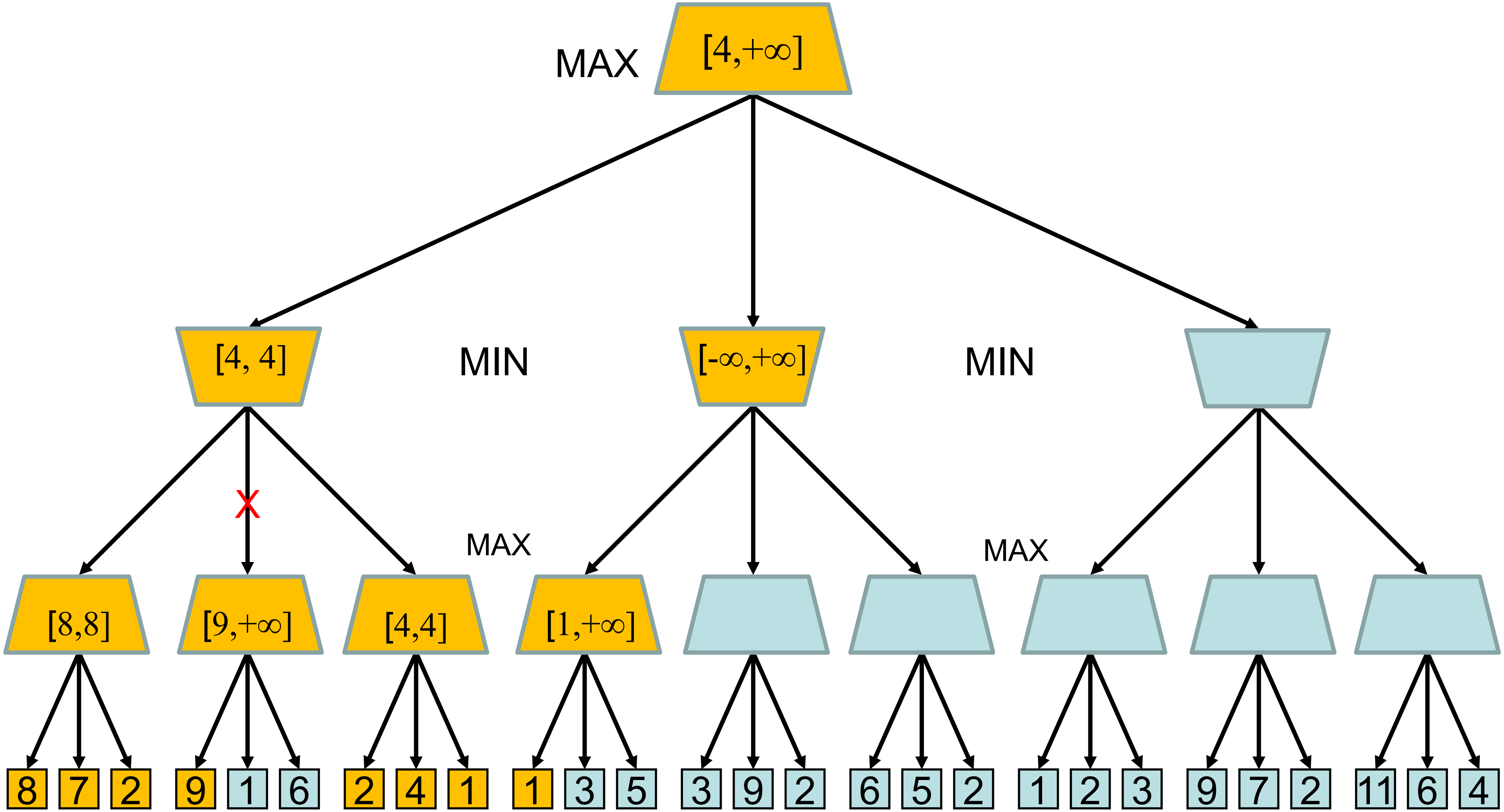
Alpha-Beta Pruning Example



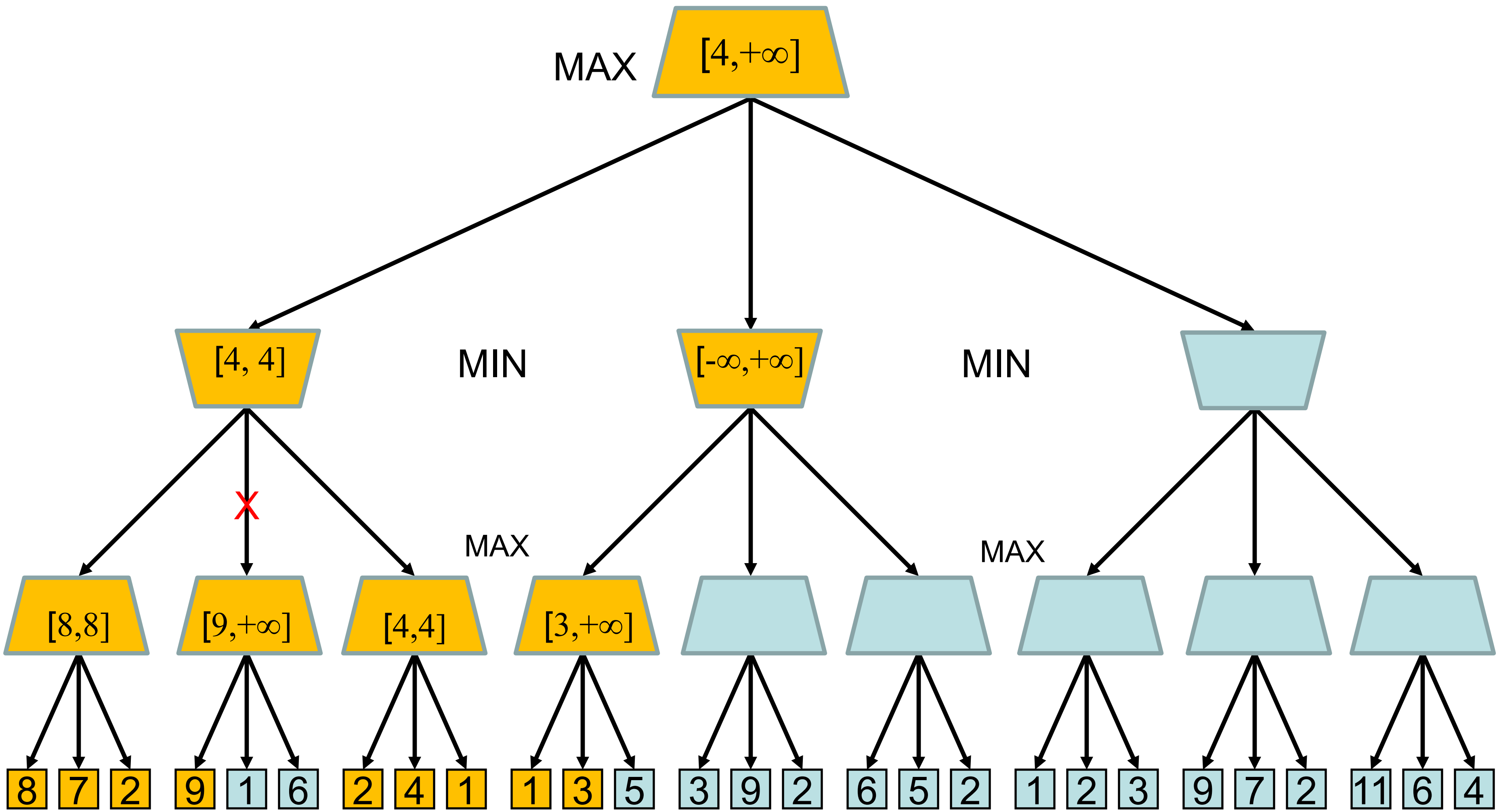
Alpha-Beta Pruning Example



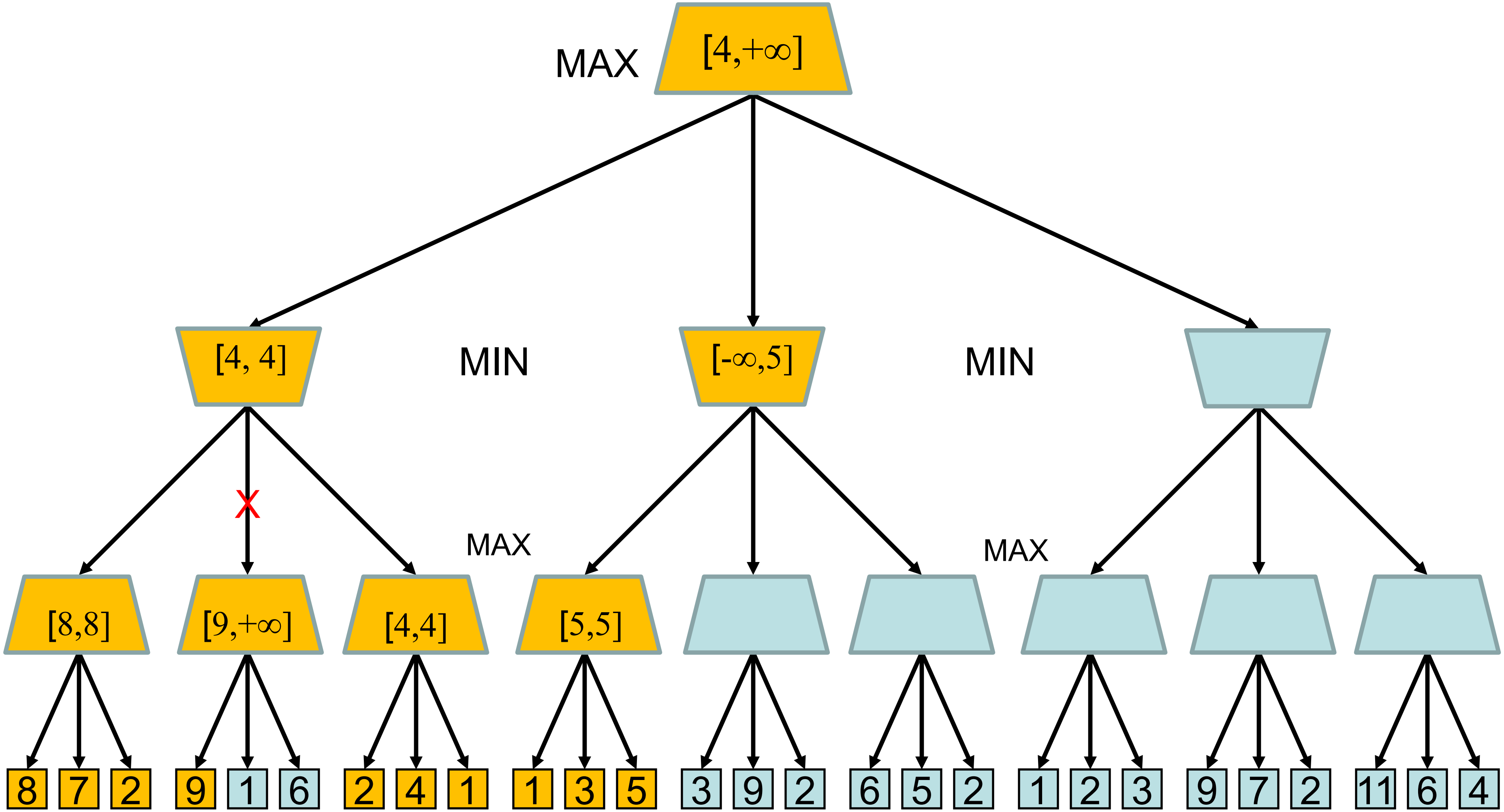
Alpha-Beta Pruning Example



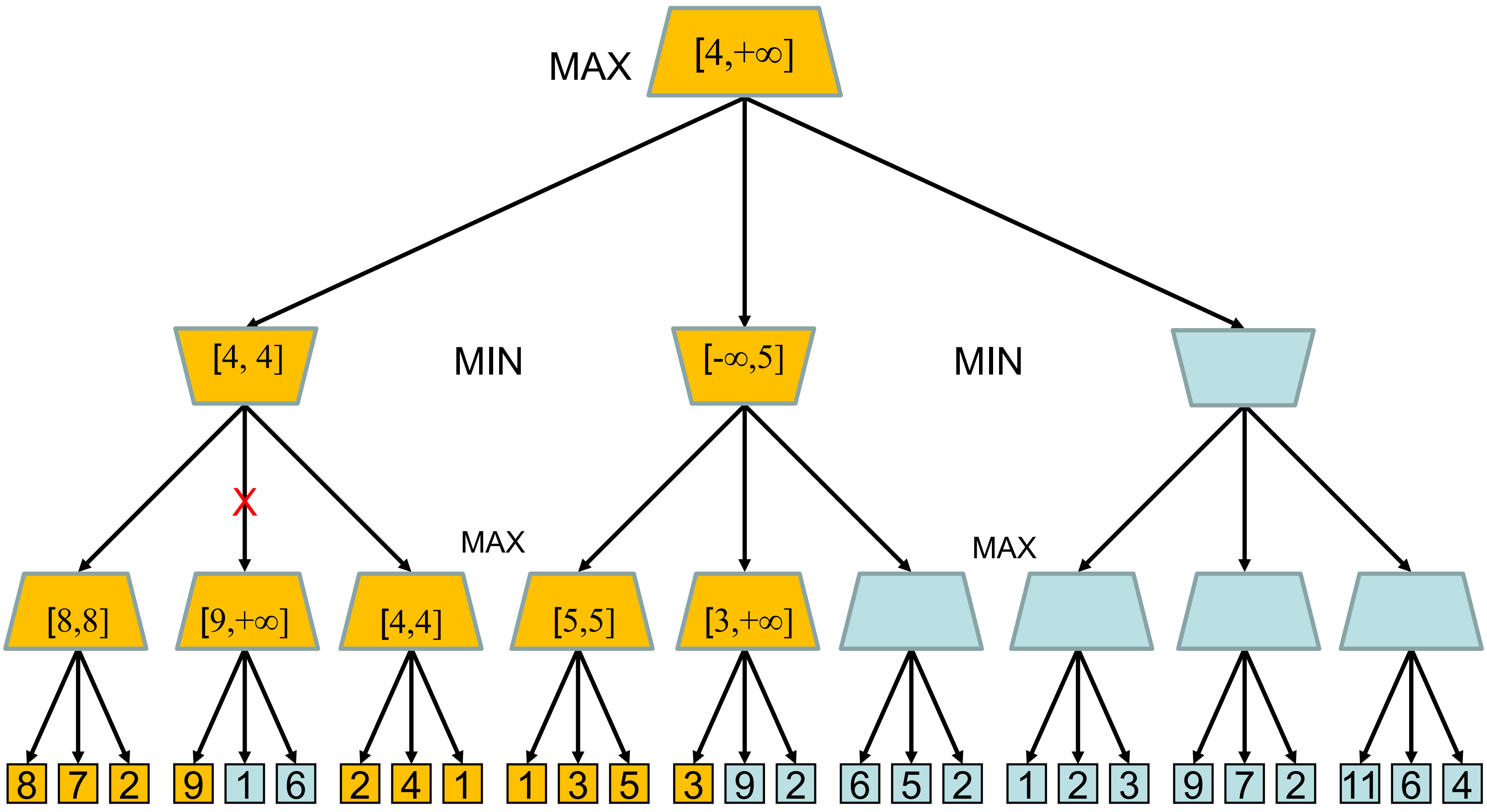
Alpha-Beta Pruning Example



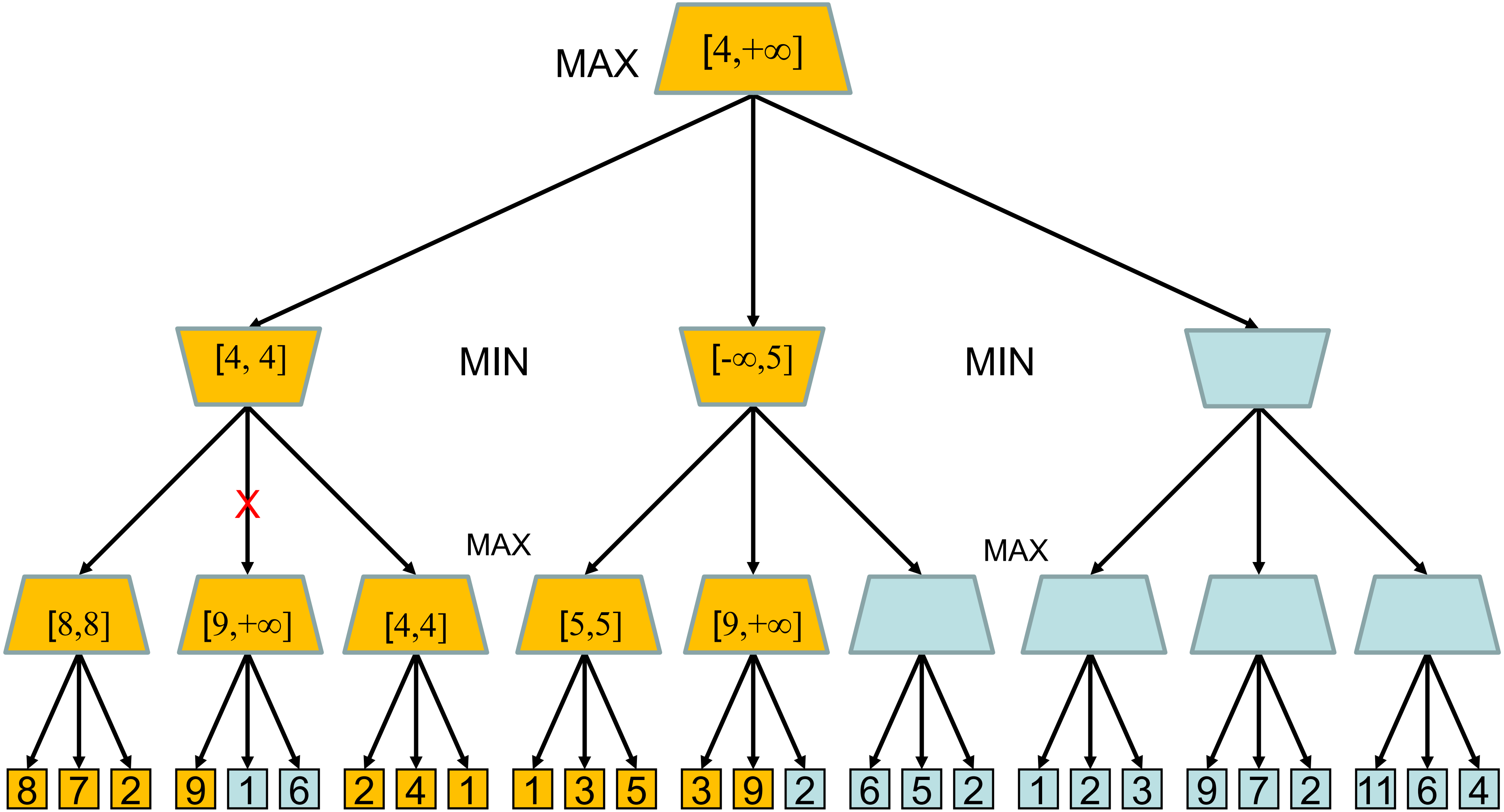
Alpha-Beta Pruning Example



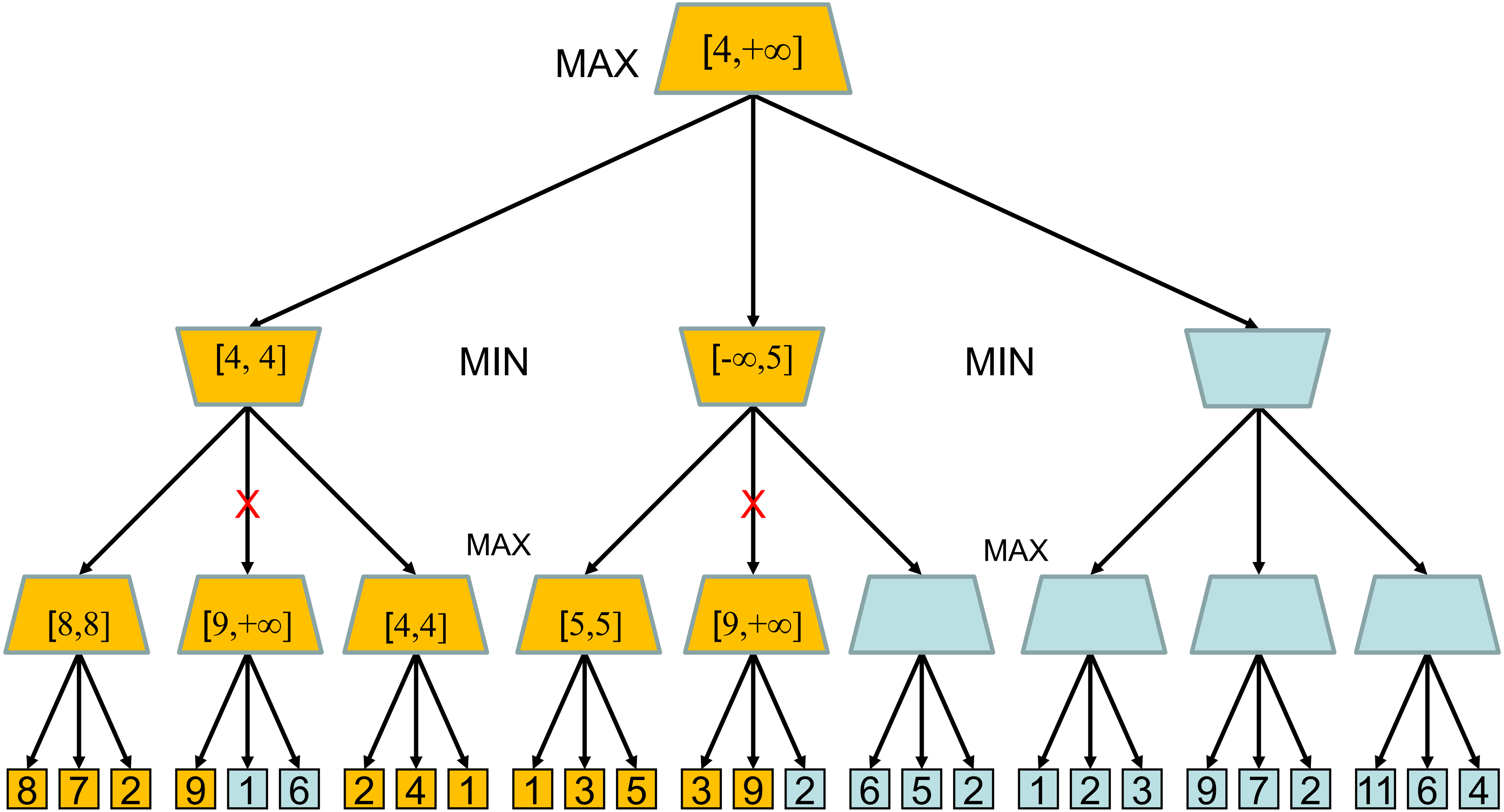
Alpha-Beta Pruning Example



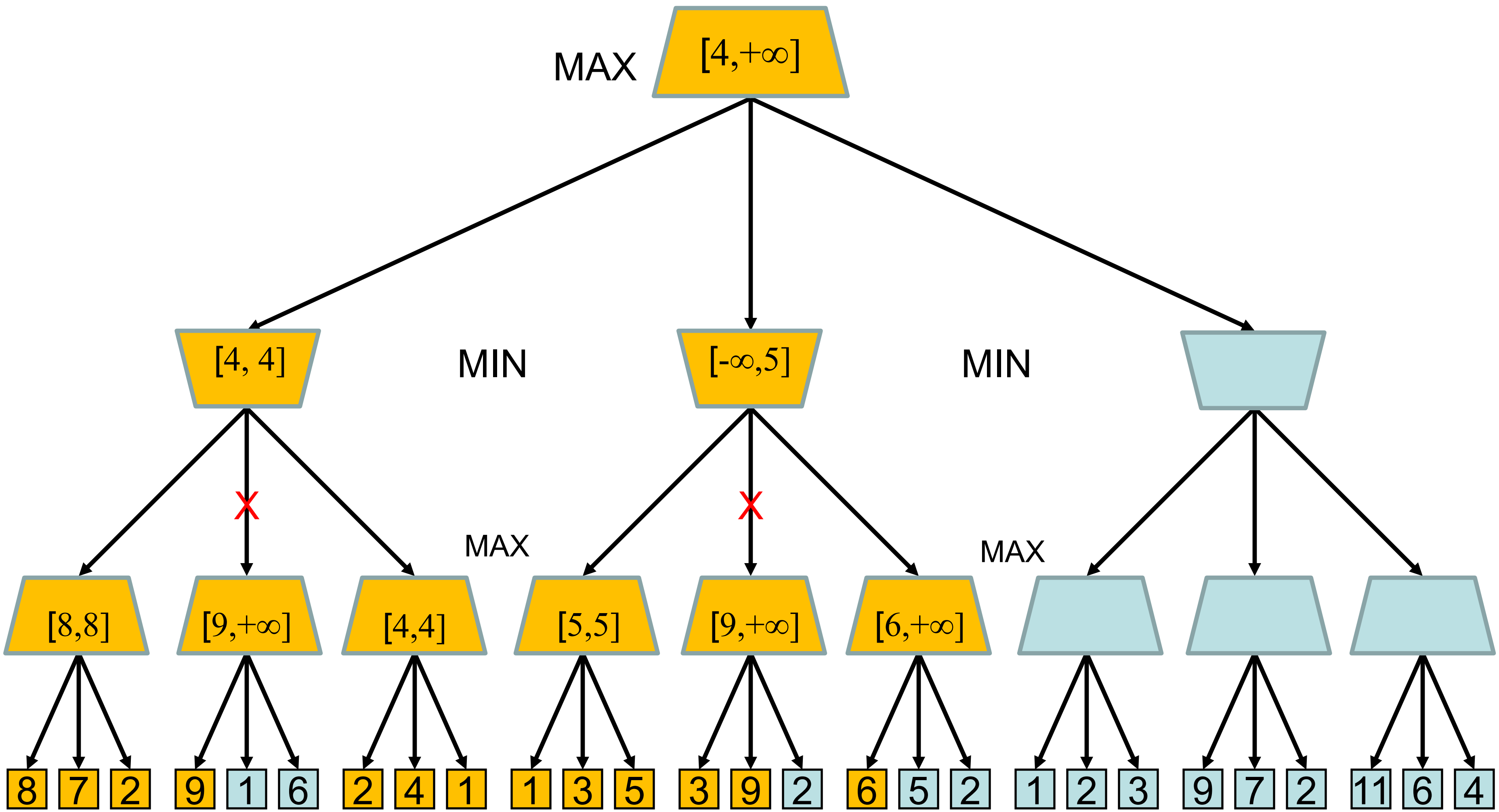
Alpha-Beta Pruning Example



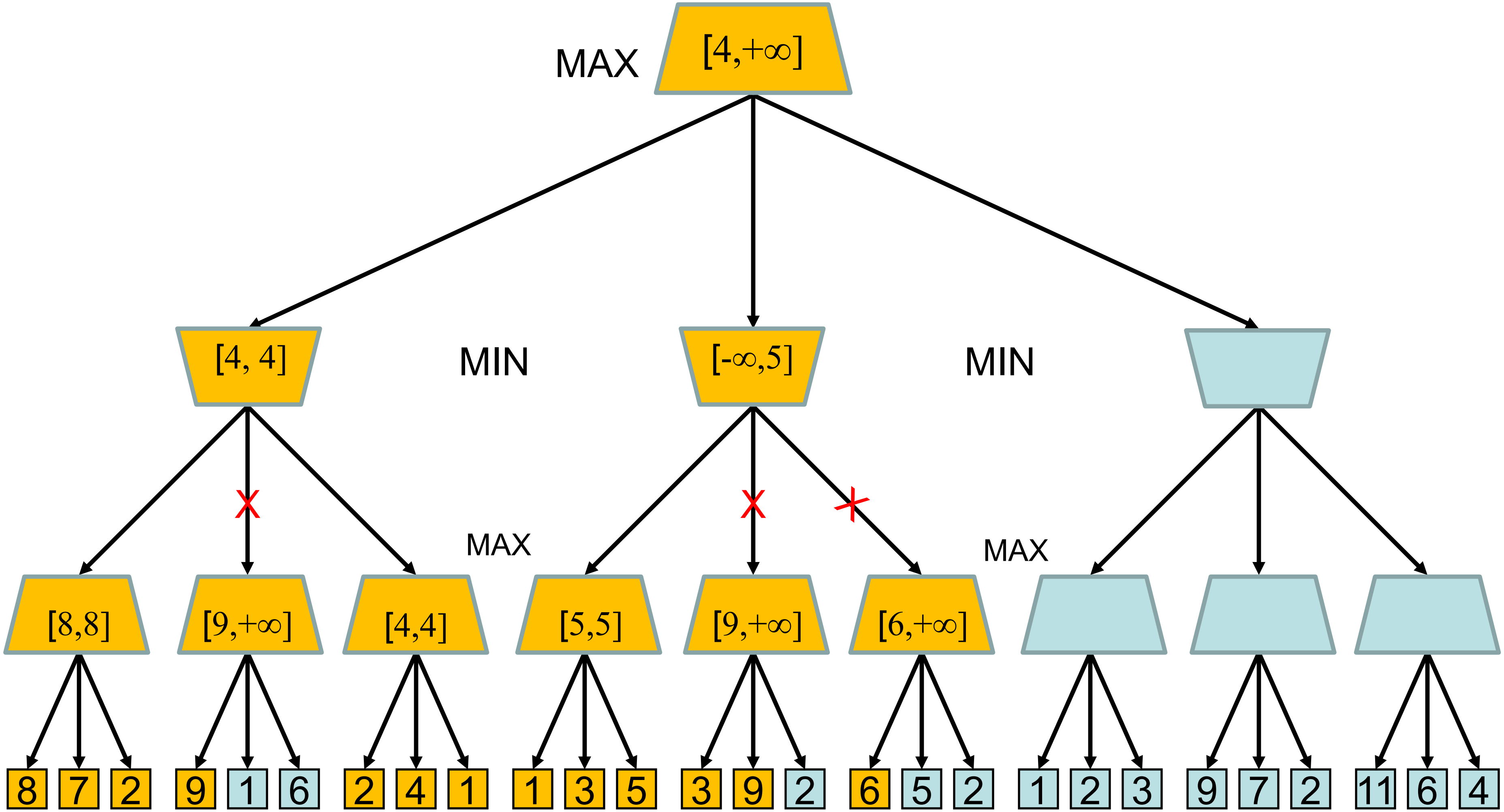
Alpha-Beta Pruning Example



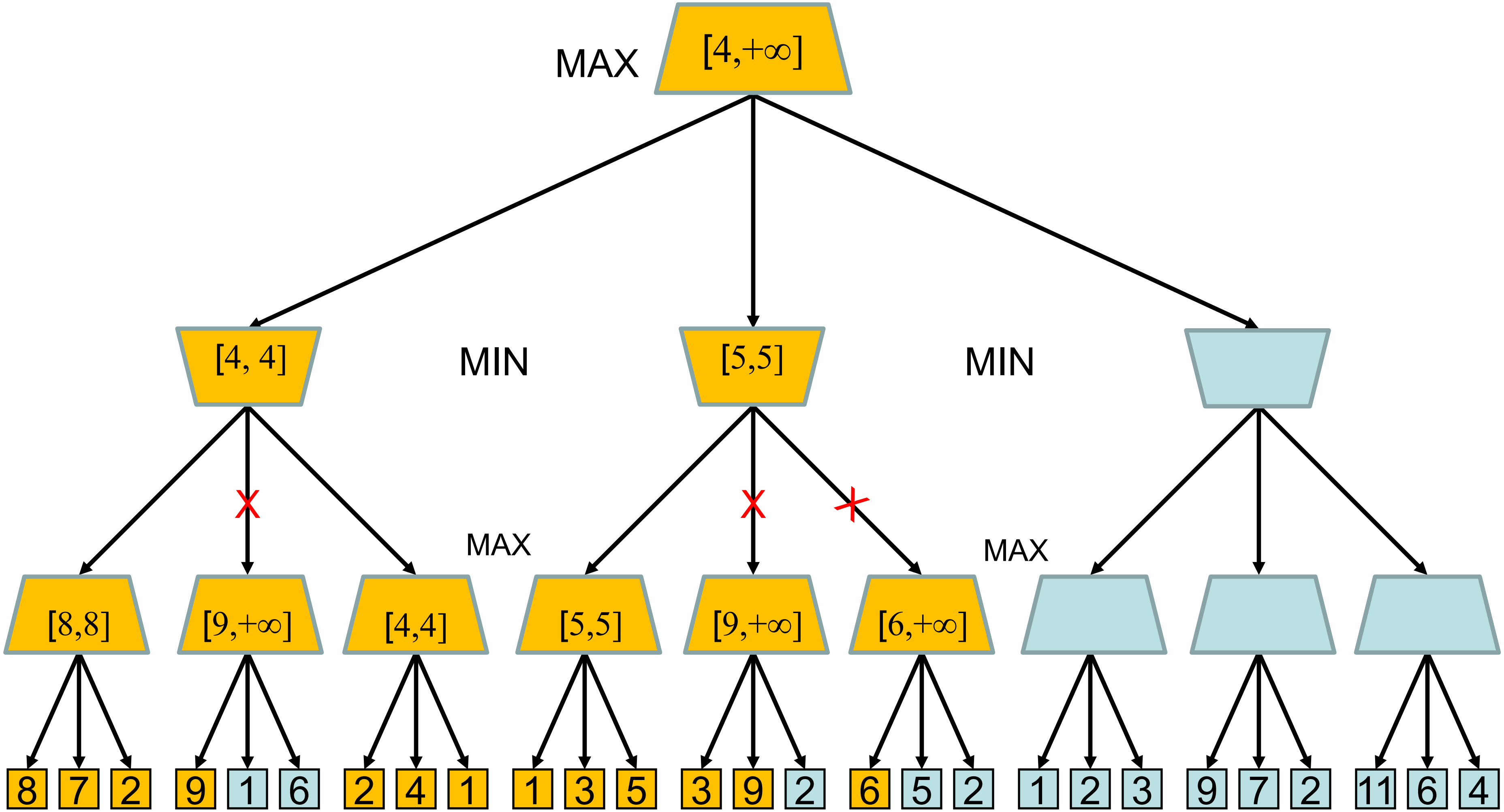
Alpha-Beta Pruning Example



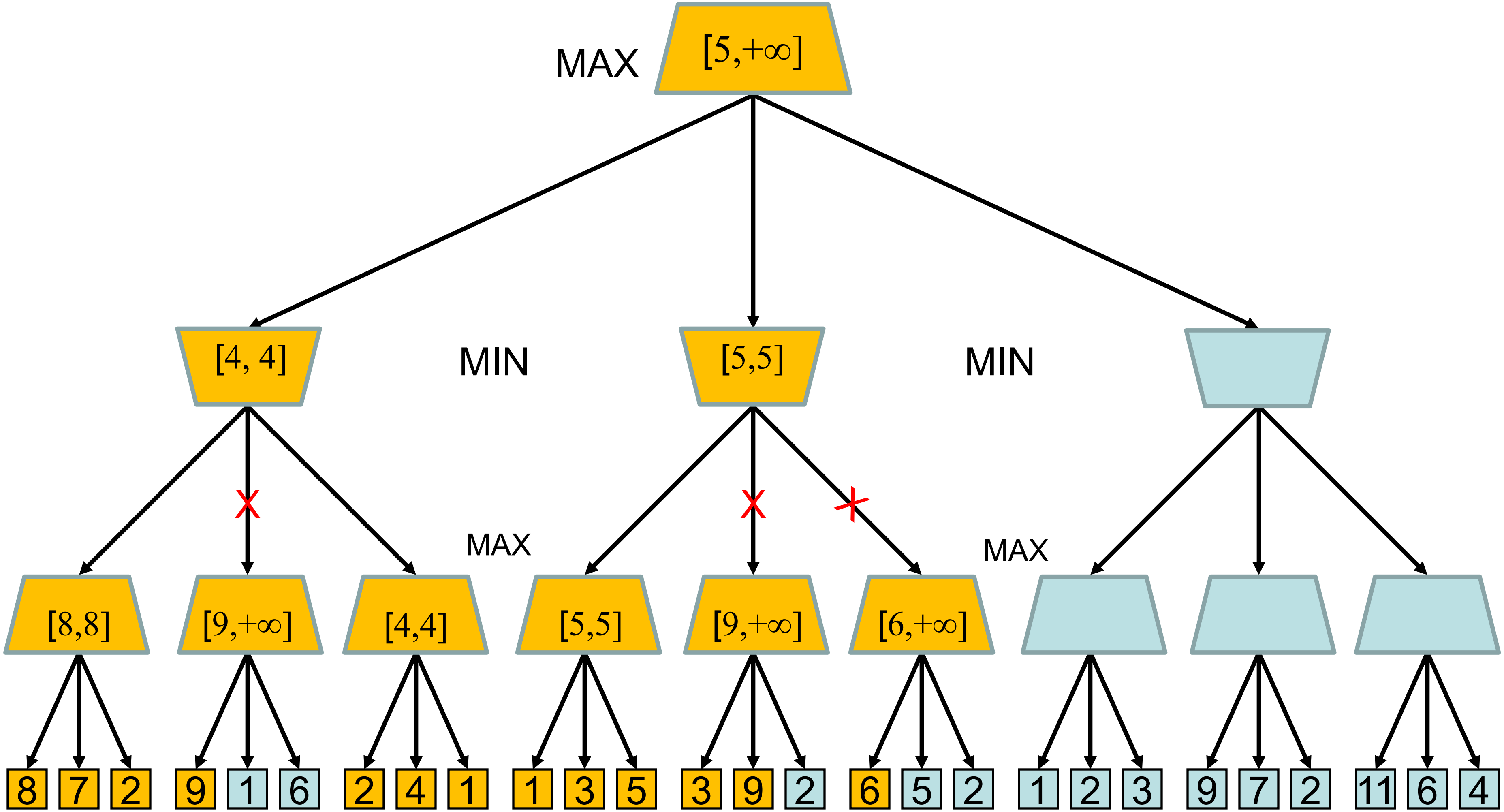
Alpha-Beta Pruning Example



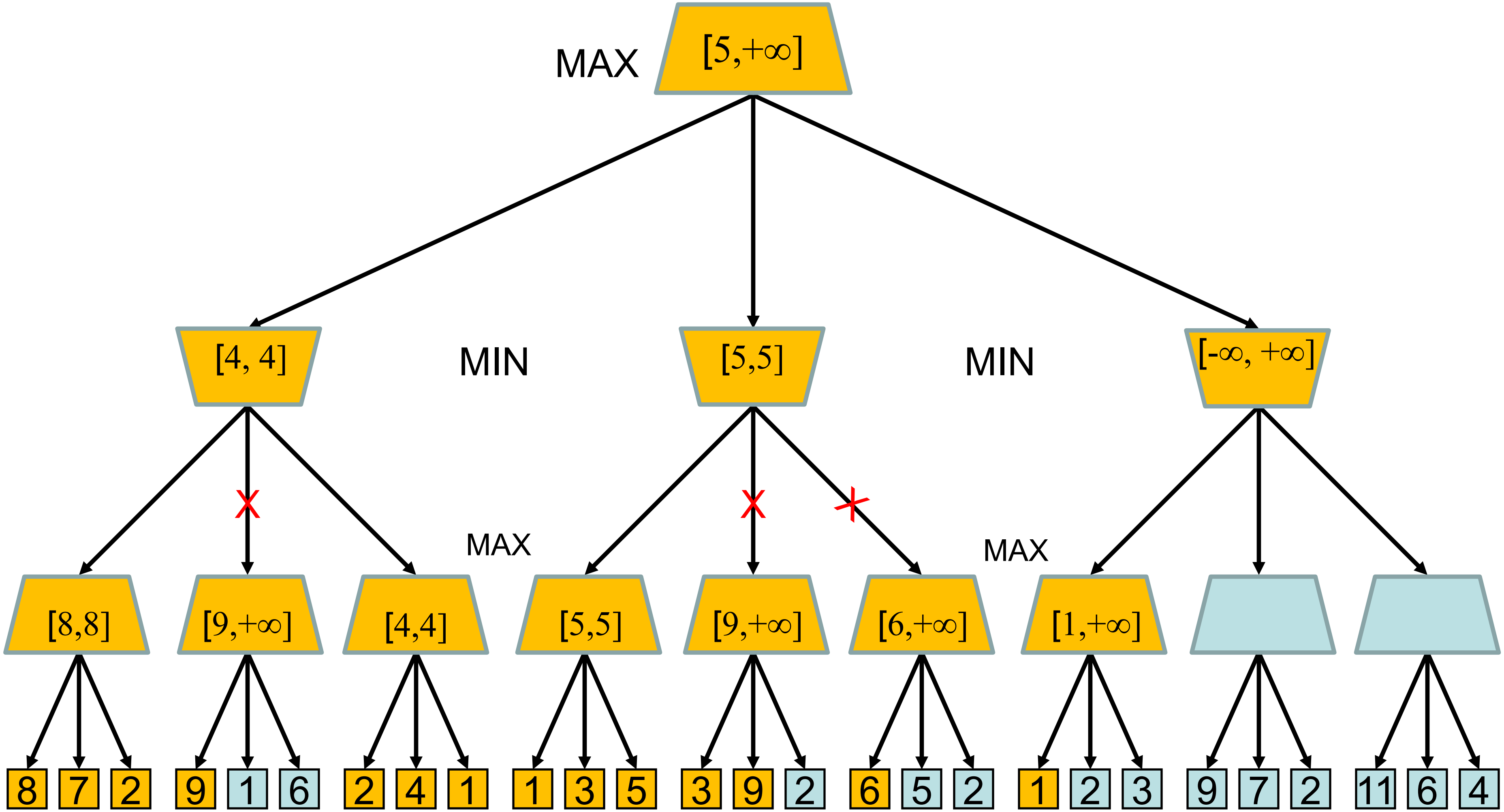
Alpha-Beta Pruning Example



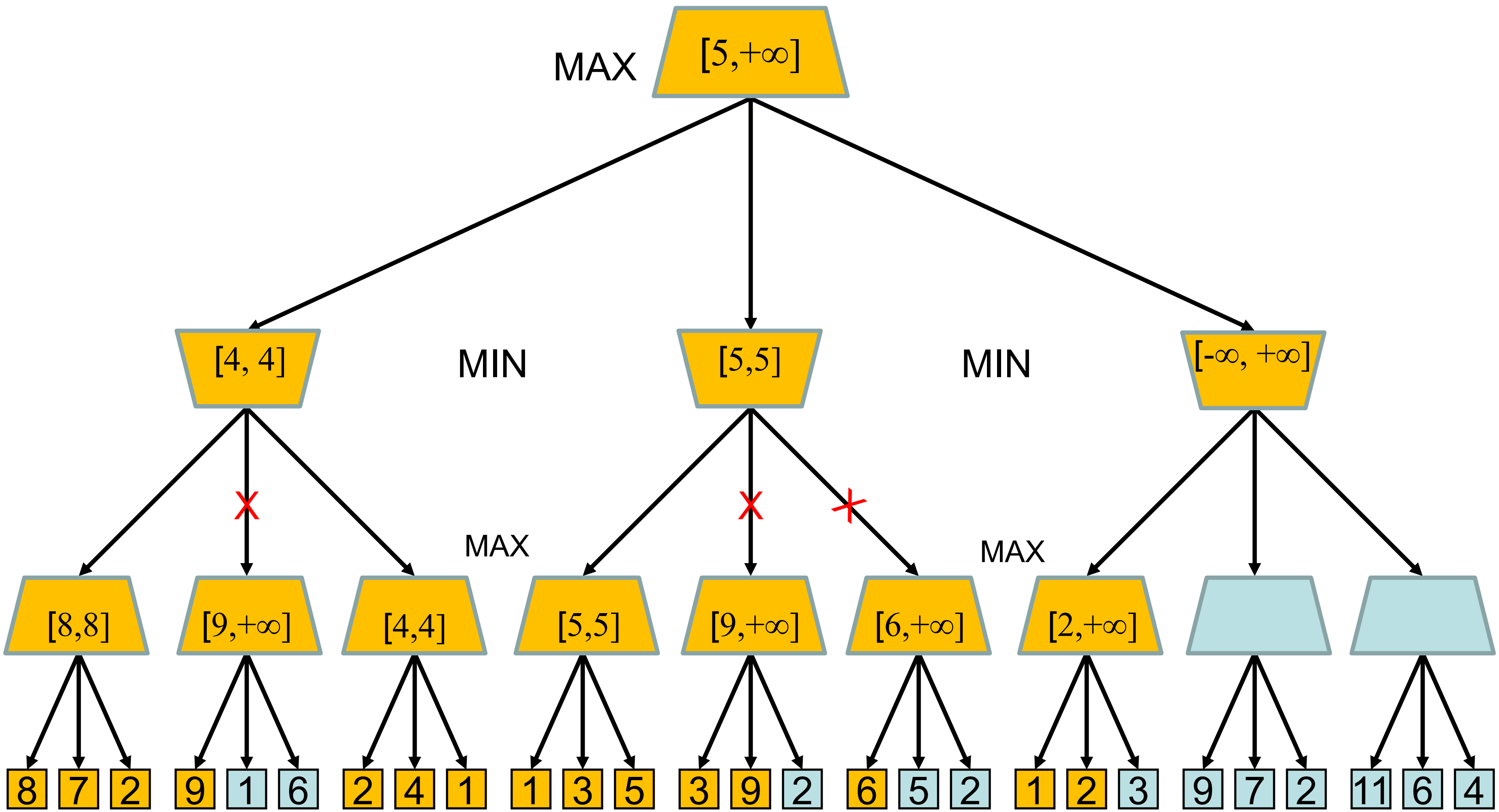
Alpha-Beta Pruning Example



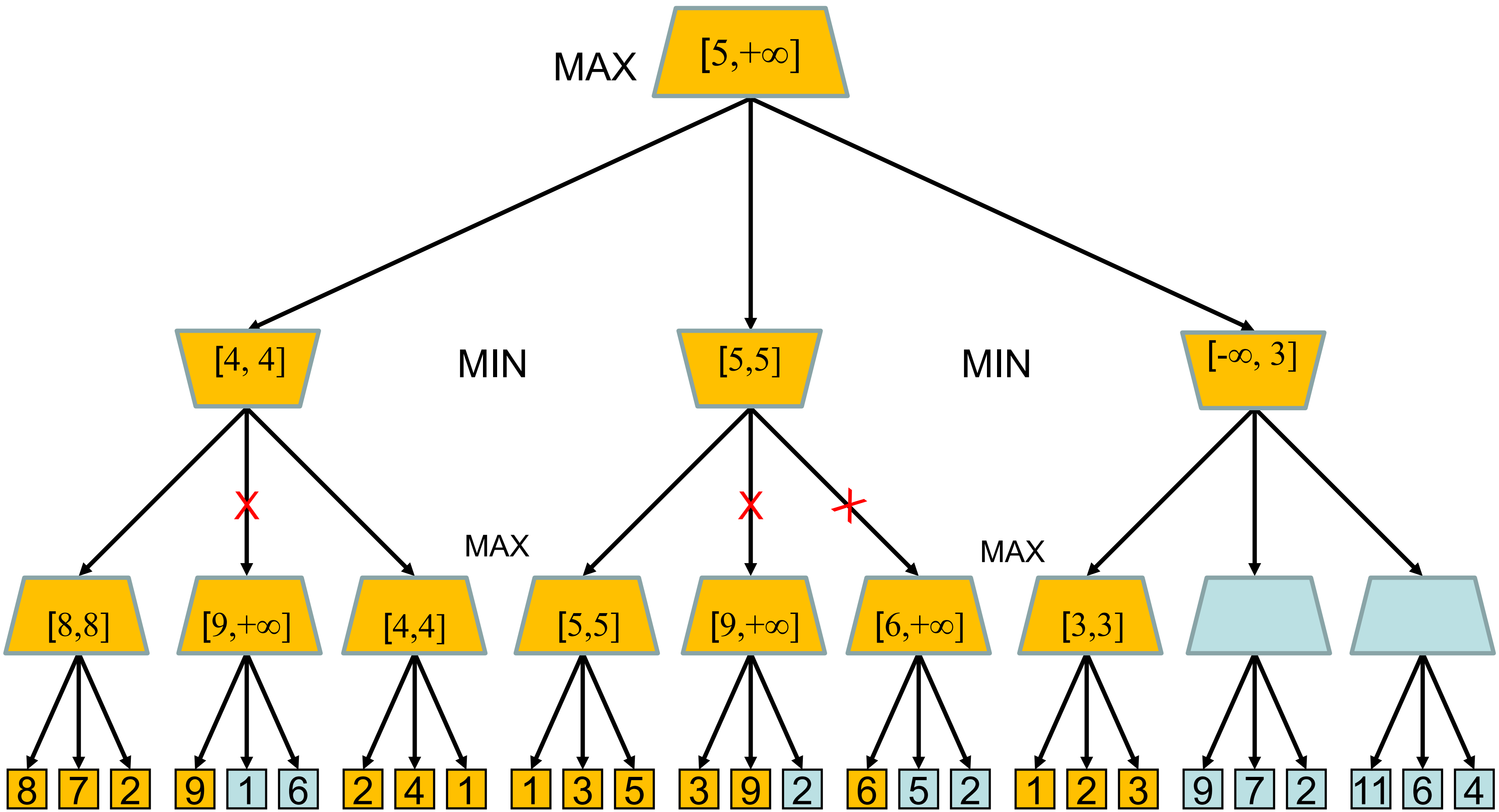
Alpha-Beta Pruning Example



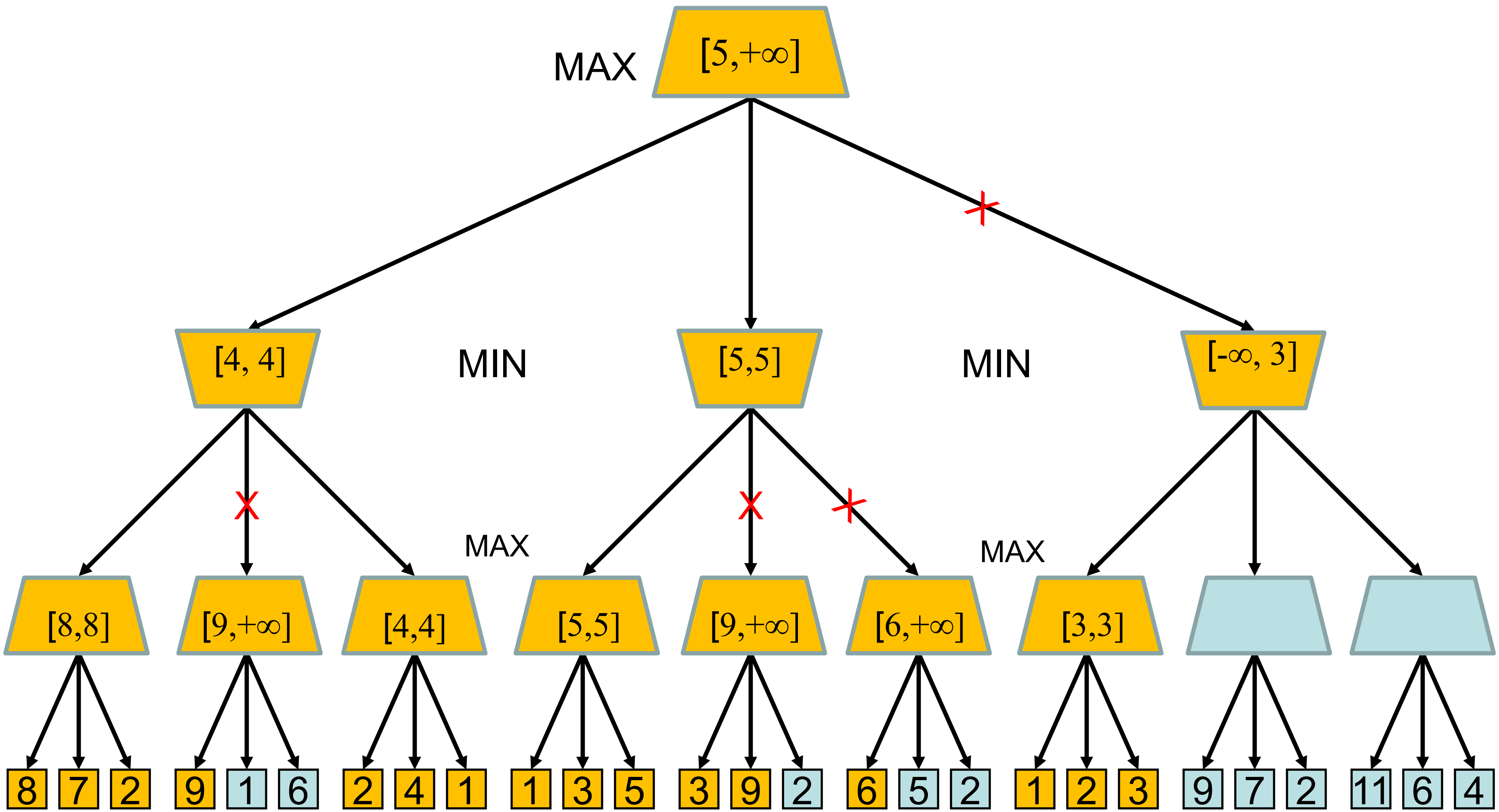
Alpha-Beta Pruning Example



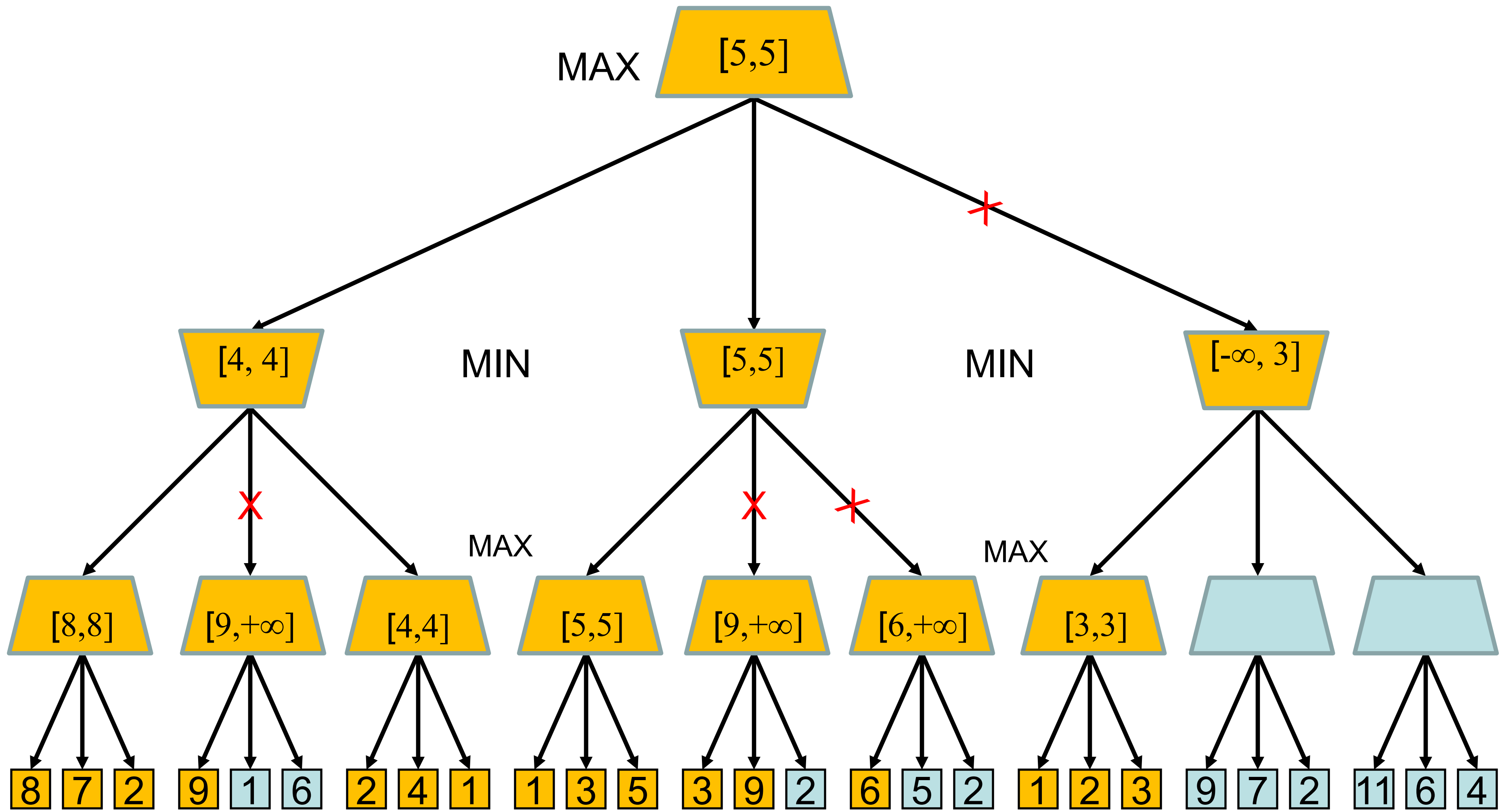
Alpha-Beta Pruning Example



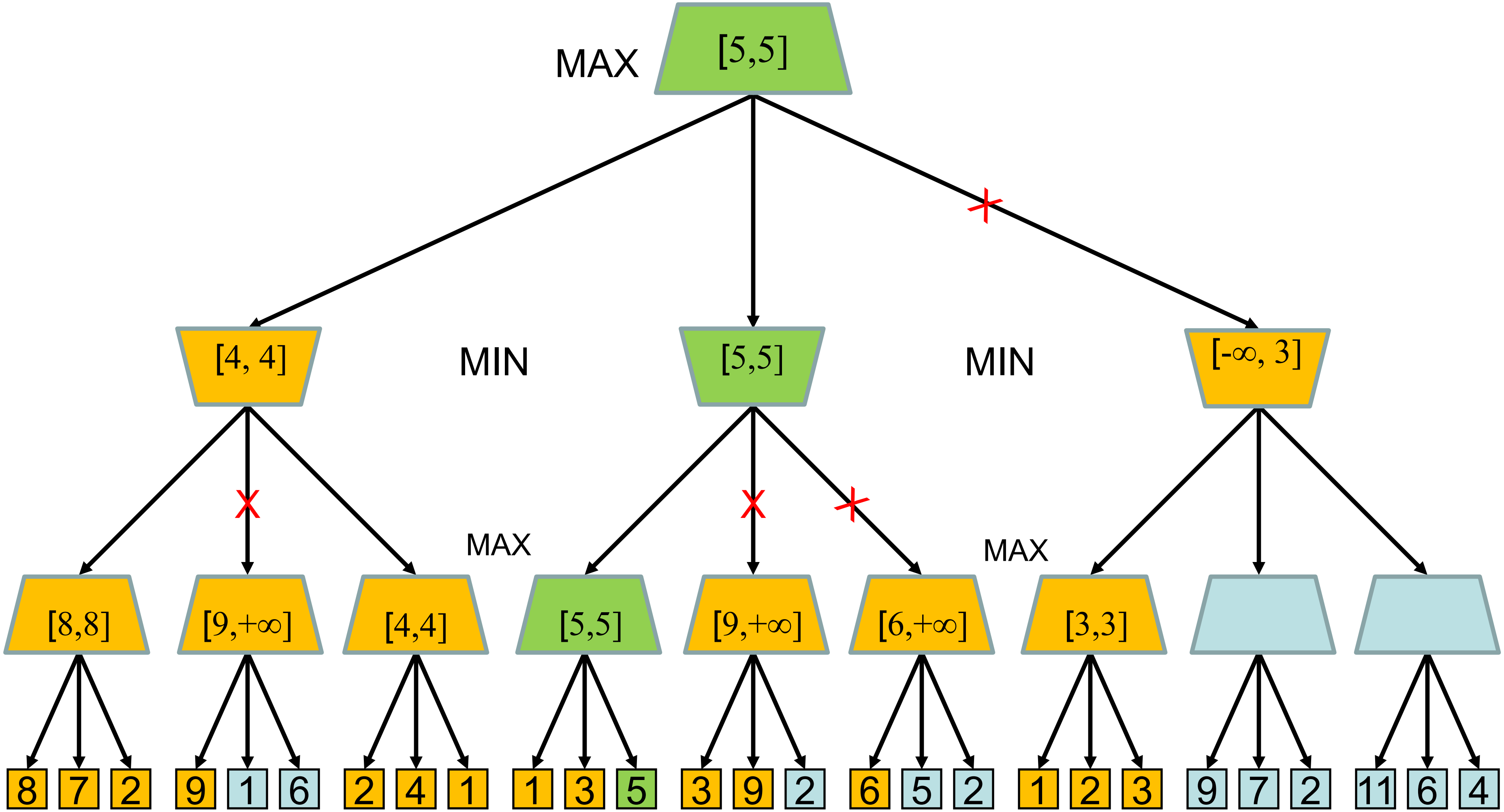
Alpha-Beta Pruning Example



Alpha-Beta Pruning Example



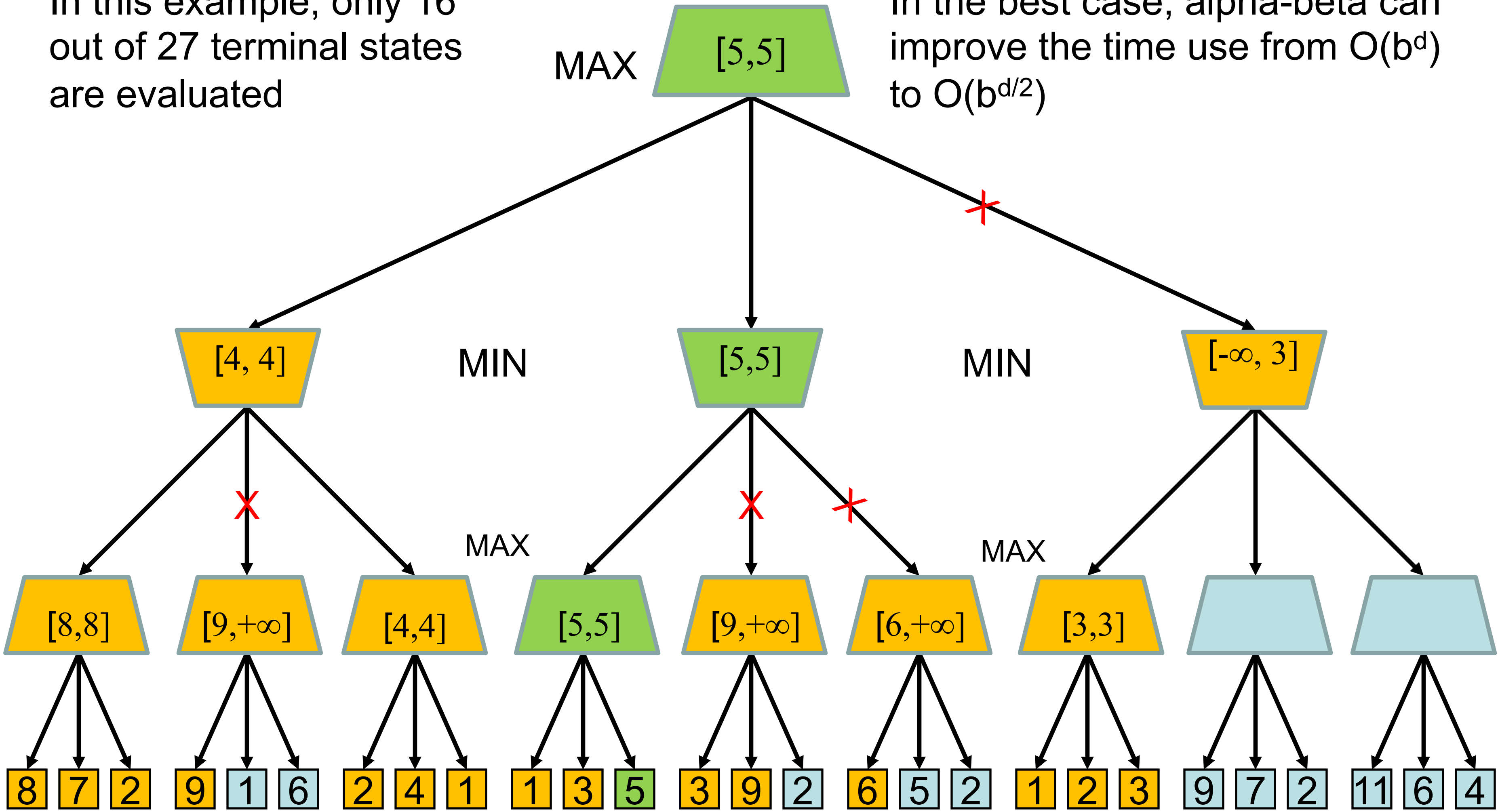
Alpha-Beta Pruning Example



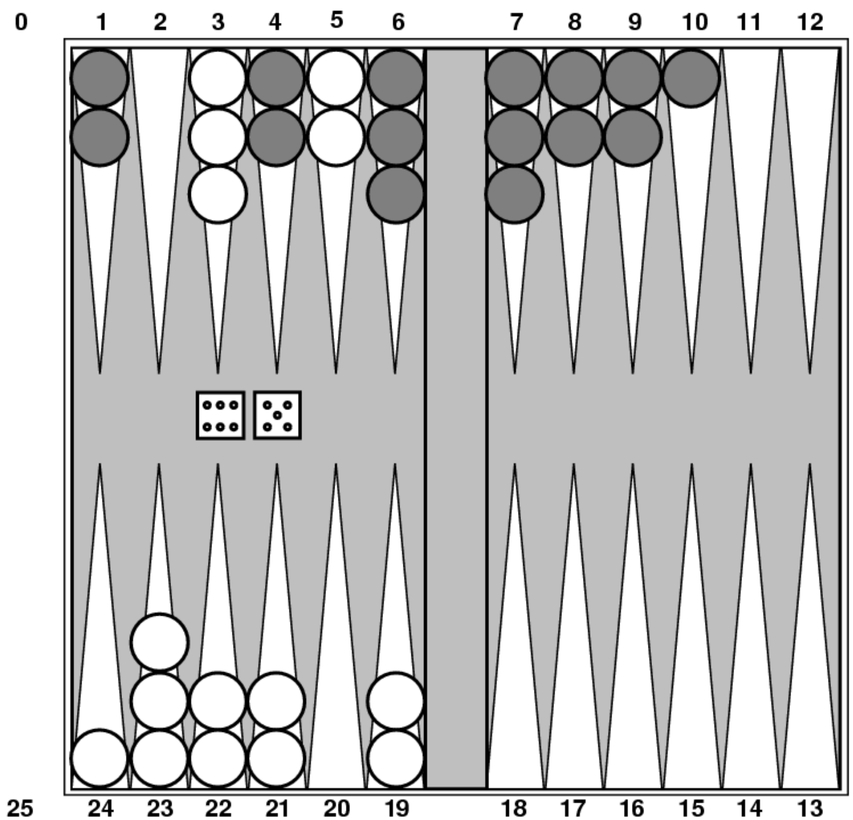
Alpha-Beta Pruning Example

In this example, only 16 out of 27 terminal states are evaluated

In the best case, alpha-beta can improve the time use from $O(b^d)$ to $O(b^{d/2})$



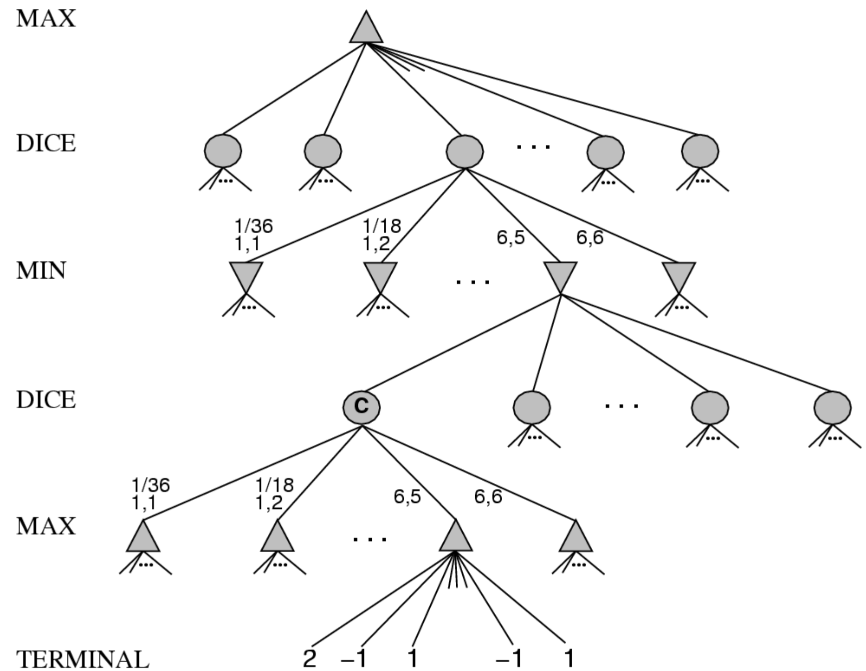
Games that Include Chance



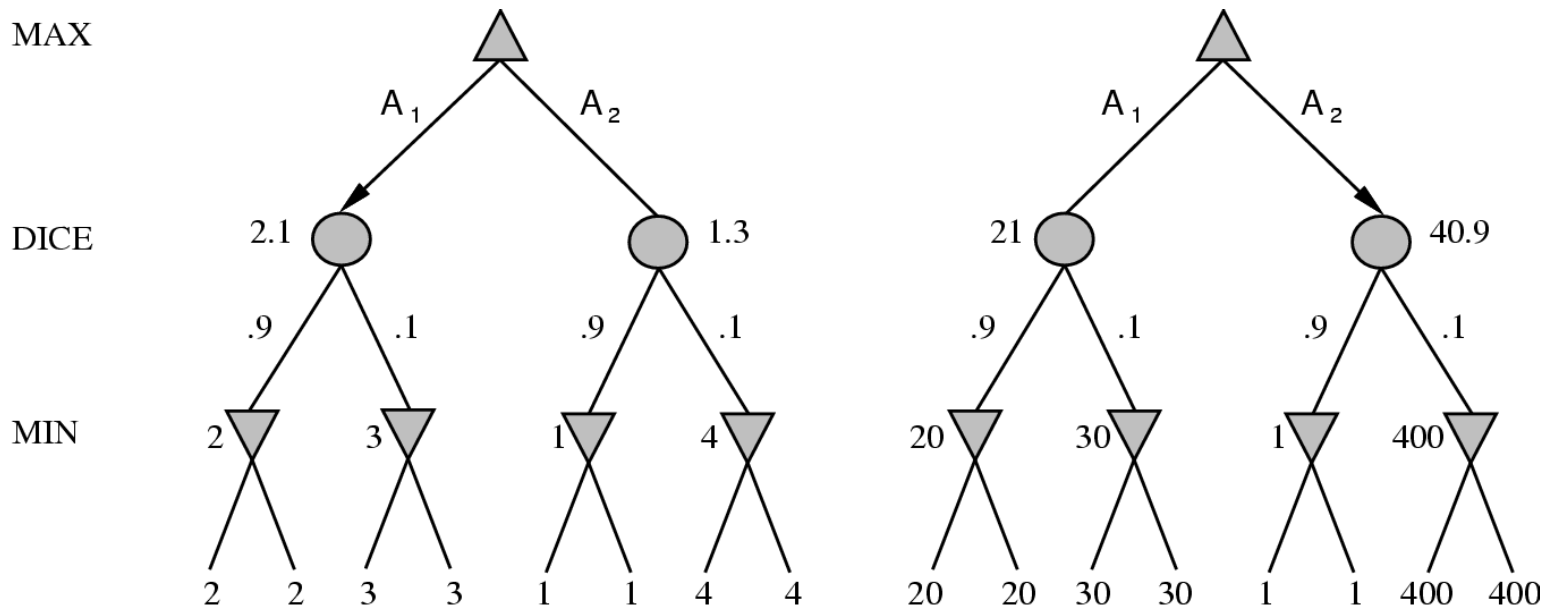
- How can we model games that involve some random chance?
- For example, dice rolls in backgammon determine the available moves
- Solution: treat the randomization as another player

Expectiminimax

- When you encounter nodes that are determined by chance, compute the expected value based on the probability distribution
- Must be careful about the evaluation functions… now these values have exact meaning rather than just ordering properties

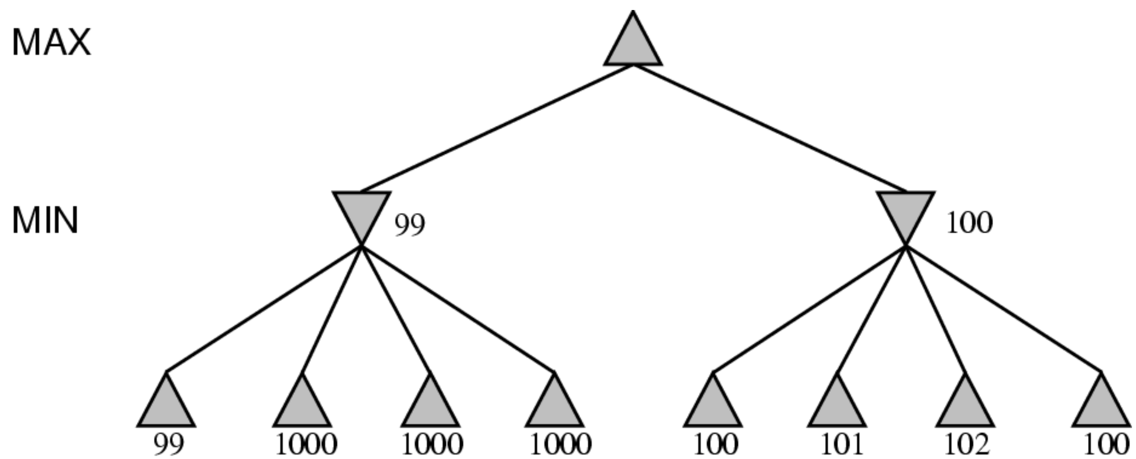


Evaluation Functions and Expectiminimax



- By changing the evaluation function values, we change the outcome.
- This would not occur under normal minimax

Is Minimax Always a Good Idea?

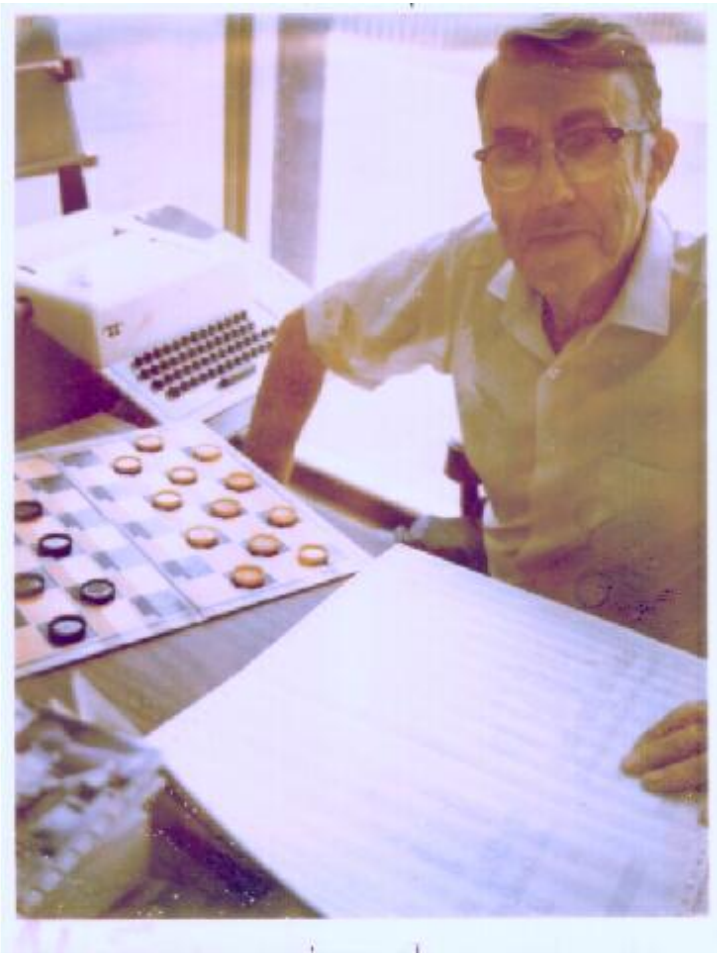


- Minimax makes the assumption that your opponent acts exactly as you would (and can look no further ahead)
- In cases like the tree above, this may be a poor assumption

Famous and State-of-the-Art Game Playing Systems

- Checkers
  - Samuel's Checkers Player
  - Chinook
- Backgammon
  - TD-Gammon
- Othello
- Chess
  - Historical Perspective
  - Deep Blue
- Go

Samuel's Checkers Player

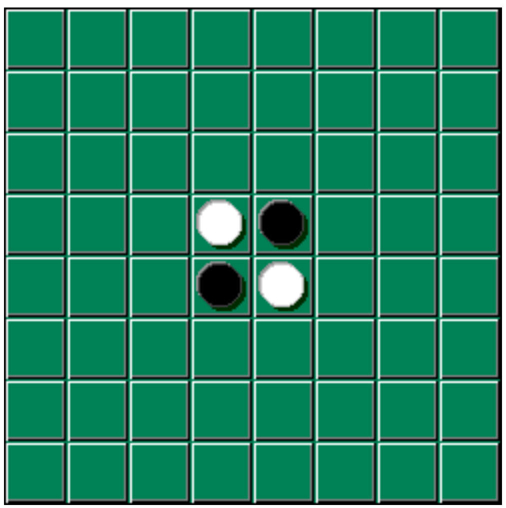


- Written in 1952
- Minimax search with alpha-beta pruning
- Evaluation function was *learned* by playing games against itself
- Played competitively after a few days of training
- Hardware:
  - 10,000 words of memory
  - Magnetic tape storage
  - .000001 GHz processor

Chinook

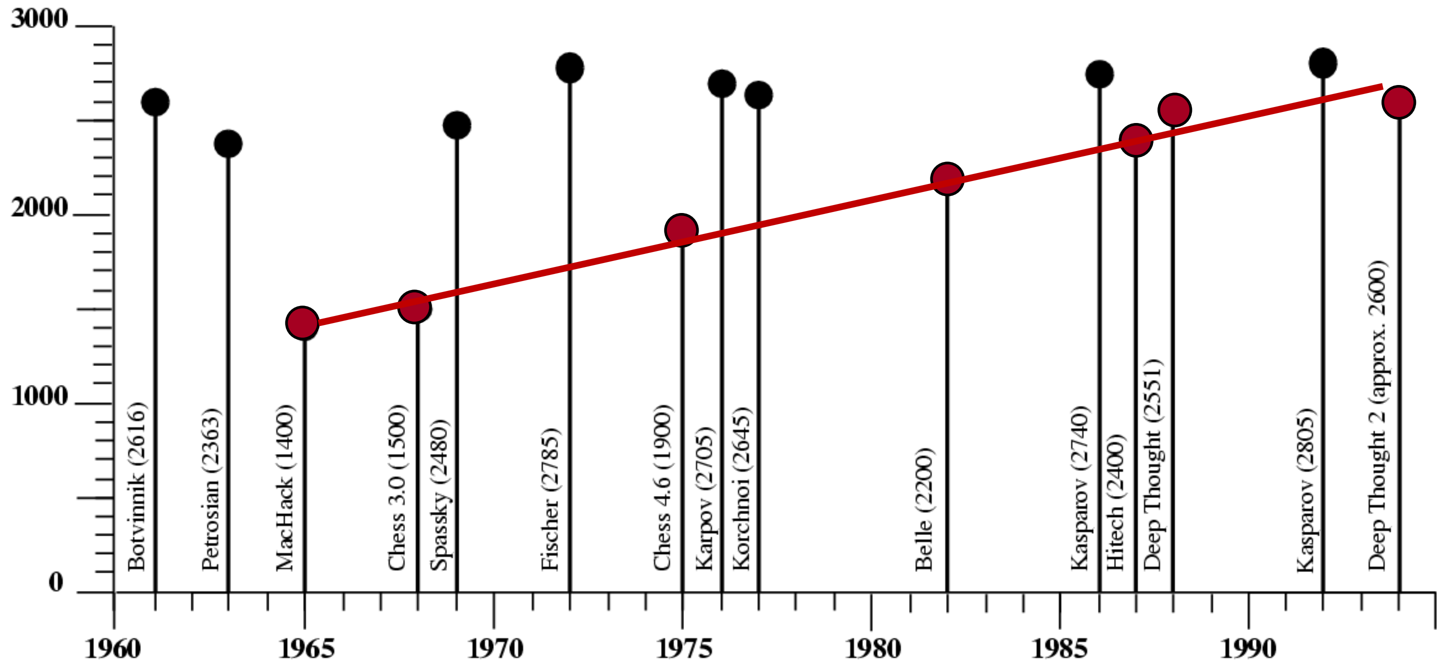
- In 1994, Chinook defeated Dr. Marion Tinsley, the world checkers champion, who withdrew from the match for health reasons
- Tinsley had held his title for 40 years, and only lost 3 matches.
- First machine to claim a human world championship title
- Incorporated end-game databases for all board positions containing 8 or fewer pieces
- Play Chinook at http://www.cs.ualberta.ca/~chinook/

Othello



- Smaller search space than chess (usually 5-15 legal moves)
- Evaluation functions are difficult to craft
- Most programs are better than human players
- In 1997, the Logistello program defeated the human world champion six games to none.

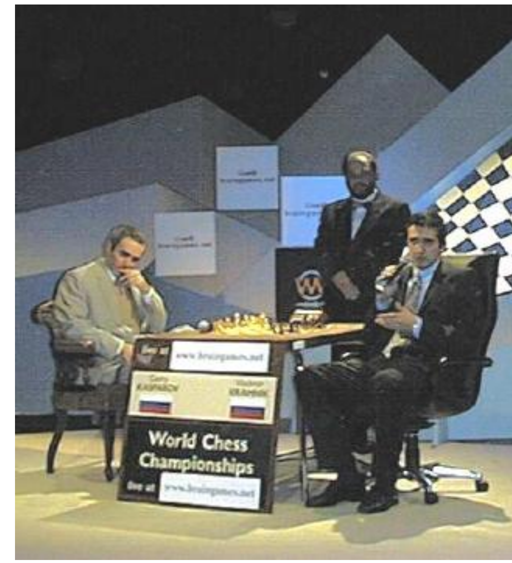
Historical Look at Chess-Playing Algorithms



- 10^{120} possible board positions
- Branching factor of ~35
- Computer chess players were increasing at roughly the rate of processor speed

Deep Blue

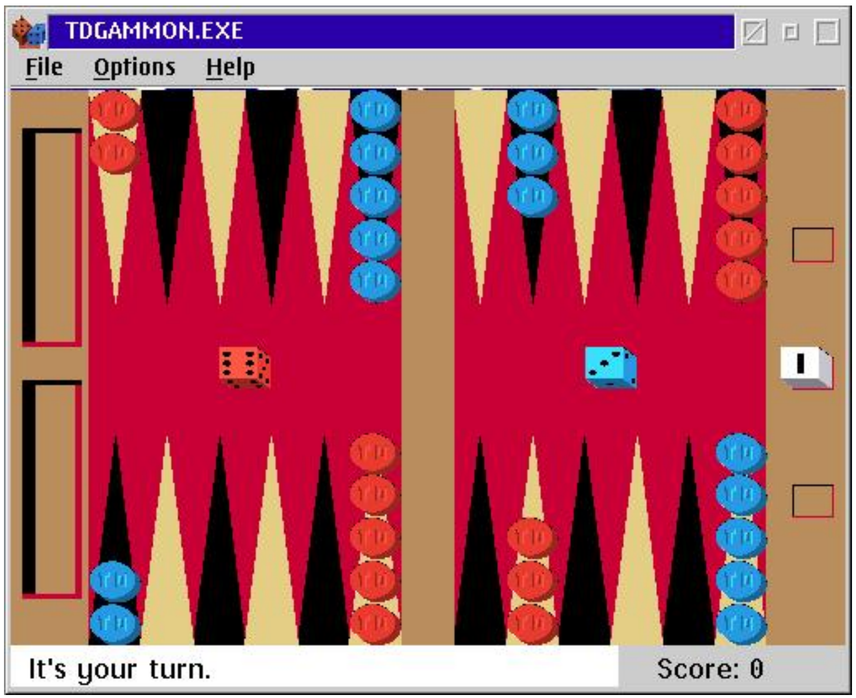
- May, 1997 defeated Garry Kasparov, the world chess champion
- Special Hardware
  - 32-node IBM RS/6000 SP high-performance computer
  - Each node contains 8 dedicated VLSI chess processors
- ~200 billion evaluations within three minutes, which is the time allotted to each player's move in classical chess (Kasparov can evaluate ~3 boards per second)
- Finely crafted evaluation function
- Not “AI” according to its creators



Deep Fritz

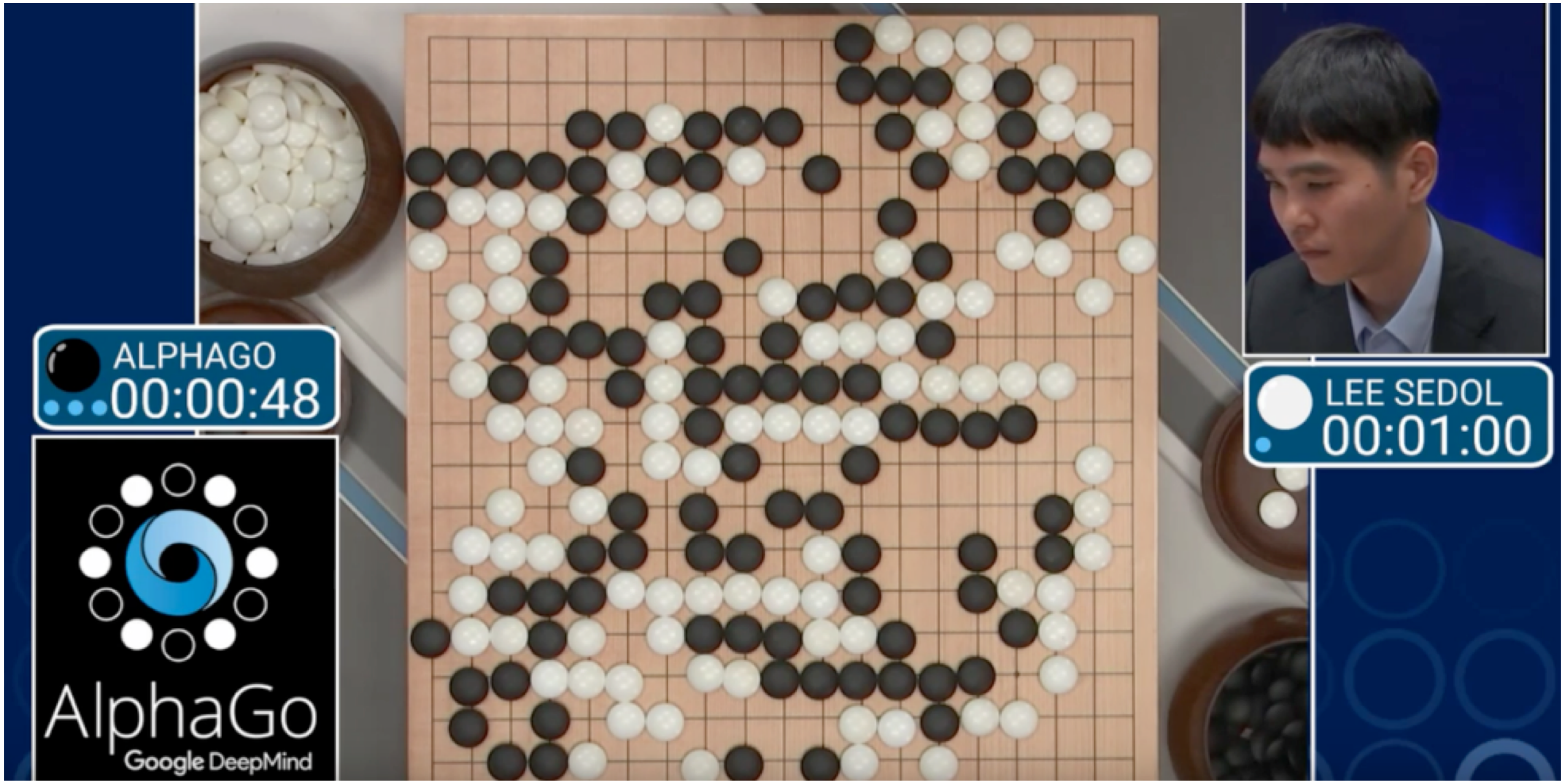
- Challenged Vladimir Kramnik (reigning world champion) in 2002.
    - (Kramnik took Kasparov's title from him in 2000)
- Eight game match ended in a draw
- Important piece:
    - FRITZ was running on an ordinary PC, not a supercomputer.

TD-Gammon (Gerry Tesauro)



- In 1998, played 100 games against world champion Malcolm Davis
- Davis won, but by a narrow margin, and mostly due to one large blunder
- Neural net evaluation function
  - 300 input values
  - 160 hidden units
  - ~50,000 weights
- 1,500,000 training matches

Google’s AlphaGo and AlphaZero

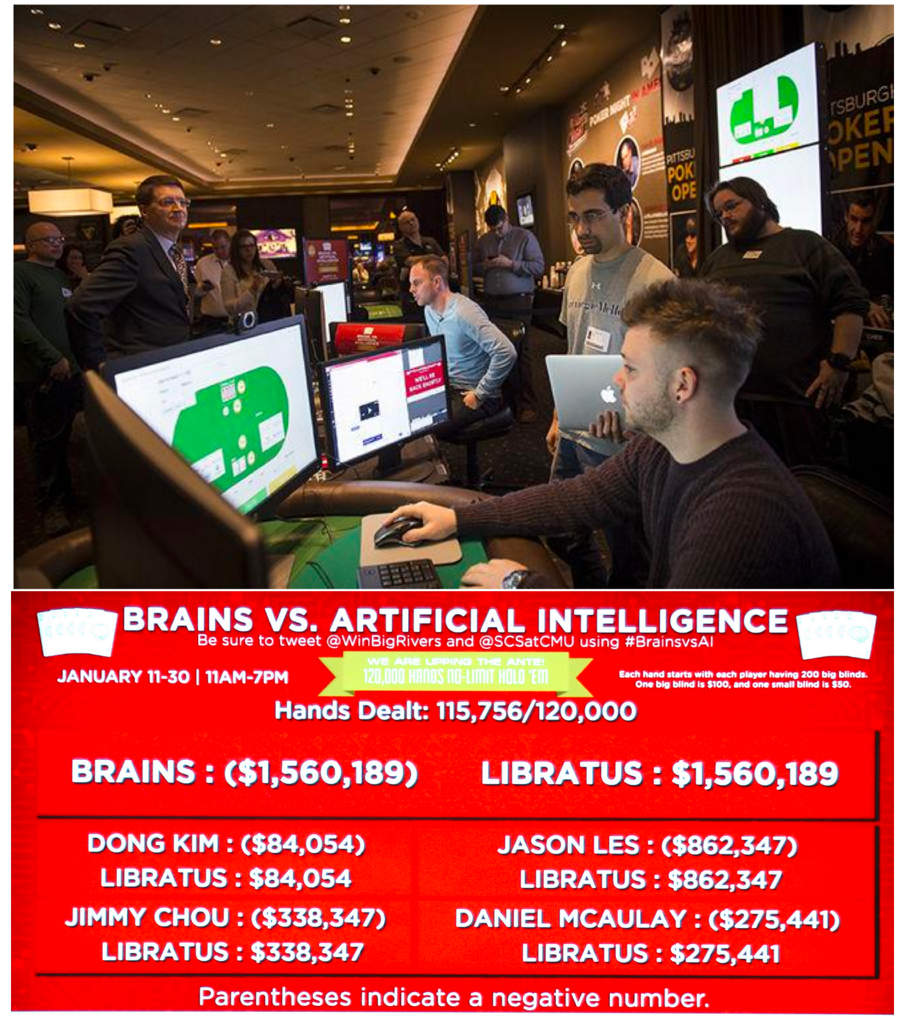


- 2016: AlphaGo beats Lee Sedol 4-1
- 2017: AlphaGo beats Ke Jie (#1 ranked human player
- 2018: Self-trained AlphaZero beats AlphaGo 100-0

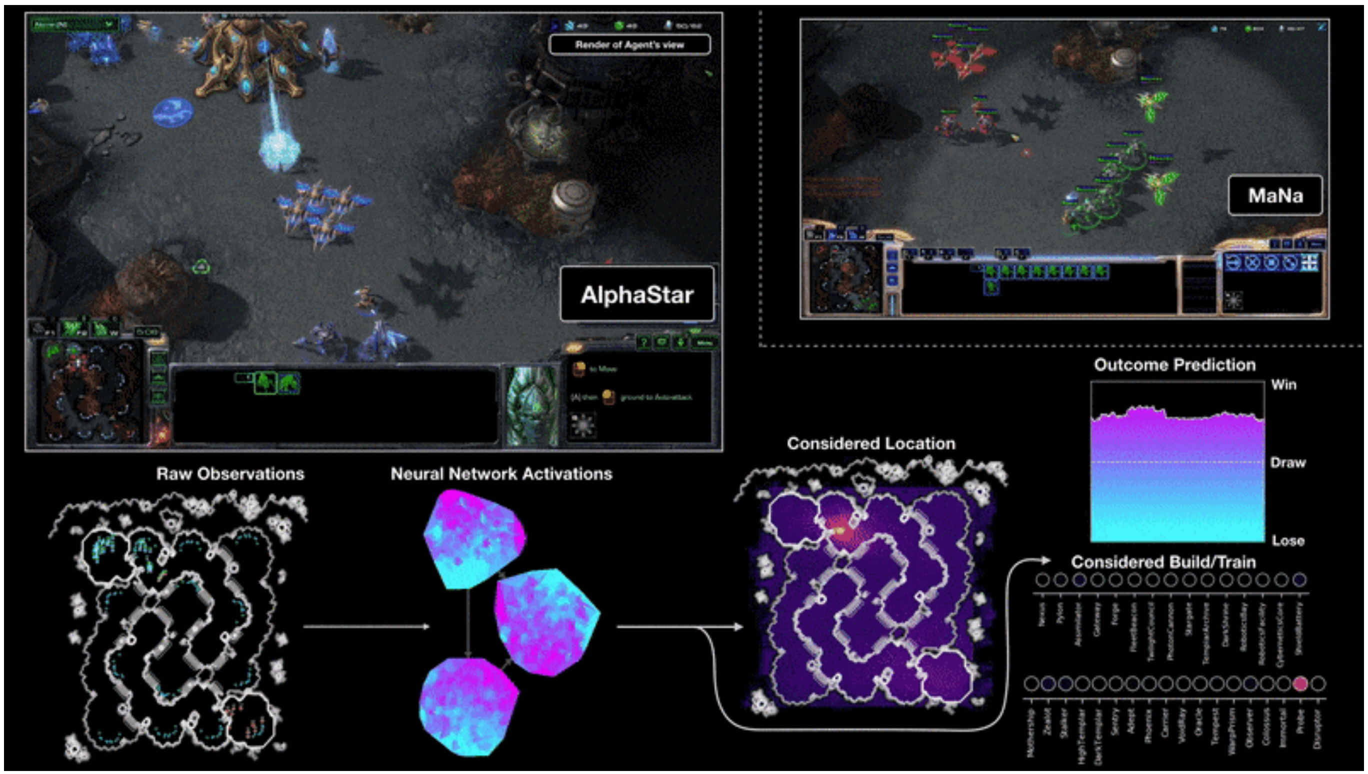
Silver, David, et al. "A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play." *Science* 362.6419 (2018): 1140-1144.

AI and Poker

- Libratus from CMU
- no-limit Texas hold-em
- 20-day match in January, 2017 against 4 top poker players
- Swapped card assignments for 2 players to eliminate luck
- Libratus was ahead $1.7m by the end of the tournament



Starcraft II



- Real-time, imperfect information, long-term planning and reward, ~100 decisions per second
- Beat 4 top human players in Dec 2018, 10 to 1
- Trained with RL and Deep networks

Relative Complexity

Game	Board Size	State-Space Complexity	Year defeated
Tic Tac Toe	9	10^{3}	1952*
Connect 4	42	10^{13}	1995*
Backgammon	28	10^{20}	1979
Chess	64	10^{47}	1997
Go (19x19)	361	10^{170}	2015
Heads up NL Holdem	N/A	10^{180}	2017
StarCraft II	N/A	10^{1685}	???

@LIV_BOEREE

Coming Up Next…

- Wednesday: no class!
- Friday: Guest lecture – Dragomir Radev on Natural Language Processing

- PS #1 due Friday
- Next week: Logical Reasoning
  - First-order logic
  - Knowledge representation