Alpha-Beta Pruning in Chess Engines

Otto Marckel

•••

Division of Science and Mathematics
University of Minnesota, Morris
April 15 2017

1997: Kasparov vs Deep Blue

https://espnfivethirtyeight.files.wordpress.com/2014/10/51659306crop.jpg?w=1824

Outline

- How Chess Engines think
- Search Trees
- Minimax
- Alpha-Beta
- Implementations in Chess Engines
- Enhancements

How Chess Engines think

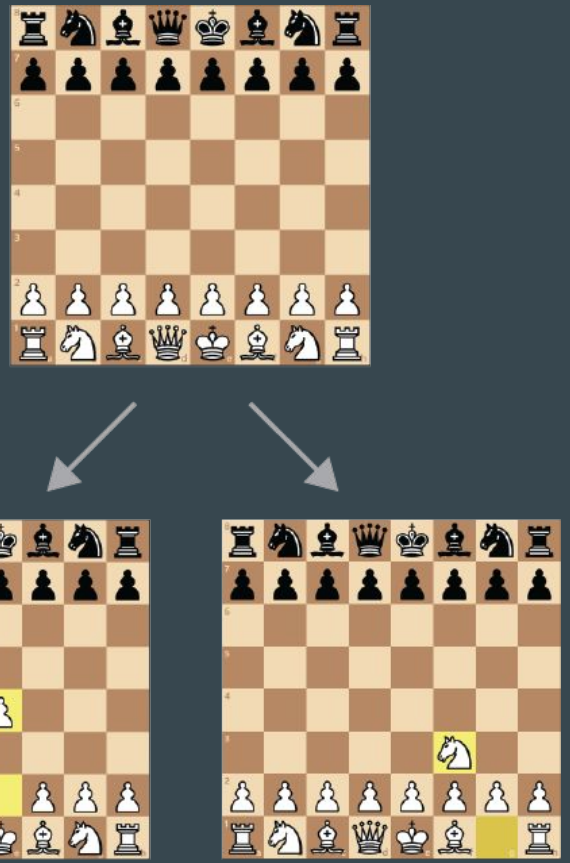
Ways to play chess

- Analysis, strategy, experience
- Possible moves
- Look at possible moves and evaluate
- Look ahead as far as we can and evaluate
- ~~Solving chess, brute force~~

Search Trees

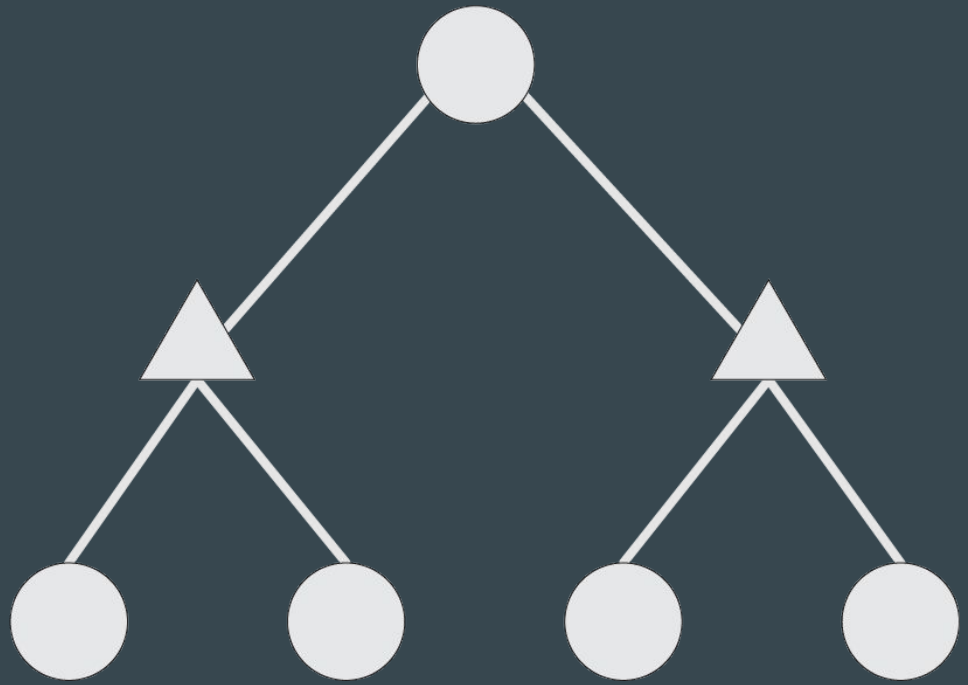
Search Tree

- The tree is created by each board state one move apart
- Each node in a branch represents alternating white and black moves
- $V = c_1f_1+c_2f_2+...+c_nf_n$
- c_i = value of one feature of the board
- f_i = function for how important the c_i is



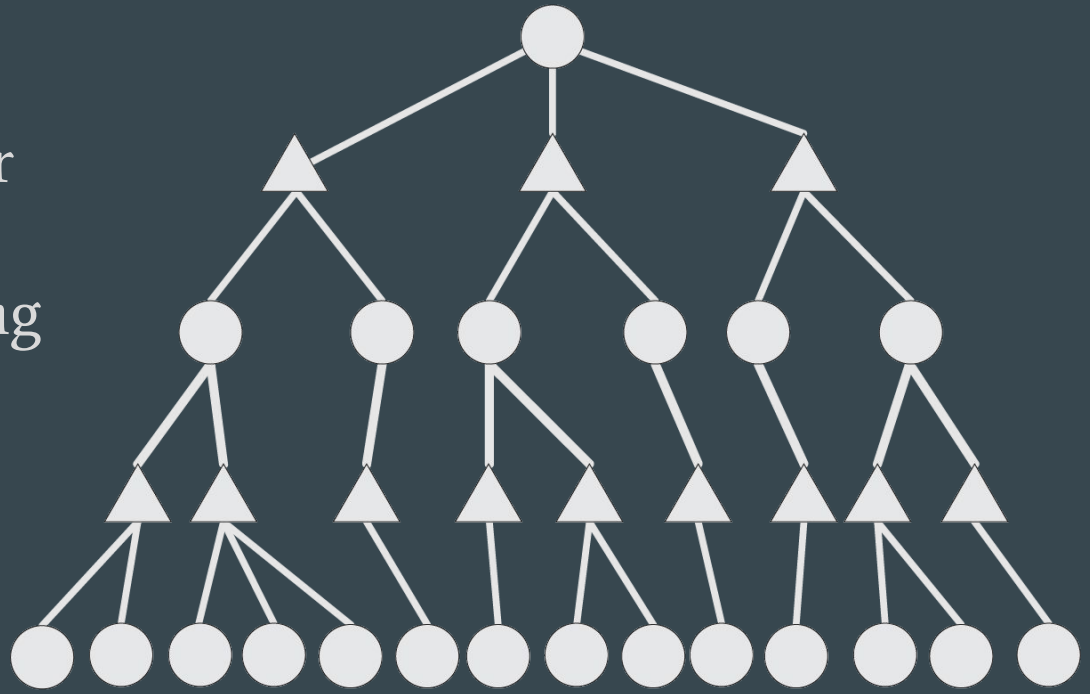
Adversarial Search Trees

- The adversarial search tree consists of nodes which captures all the possible moves in the game
- Each move is represented in terms of loss and gain for one of the players
- Move of Max player is a circle
- Move of Min player is a triangle



Adversarial Search Tree

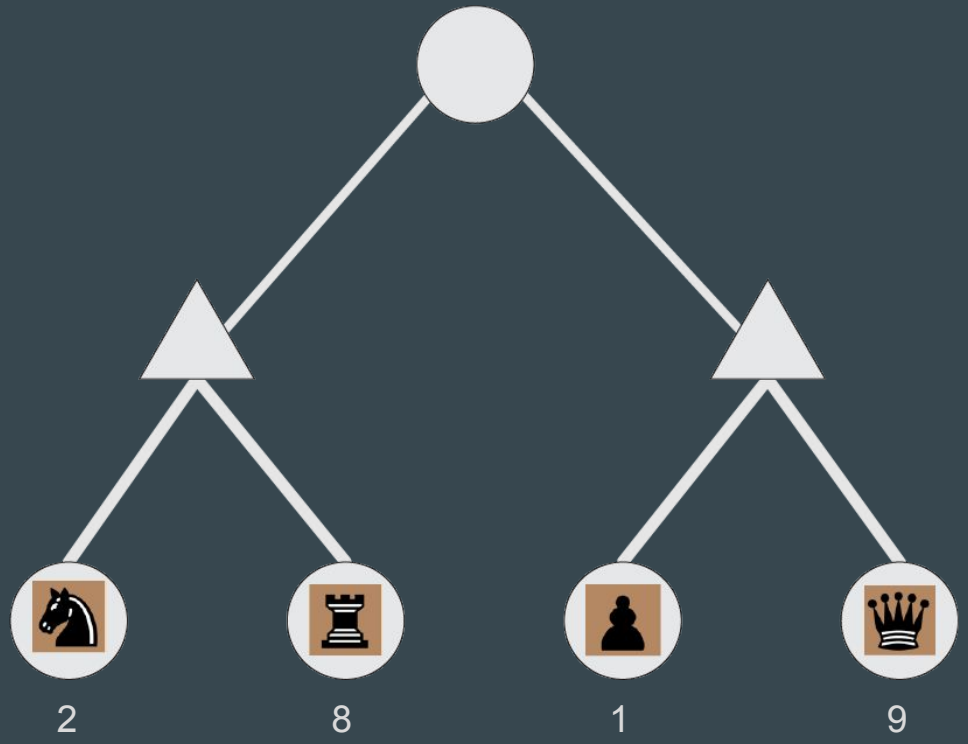
- Used for multiple player games
- Alternates who is making a move
- Chess is a two-player, zero-sum, perfect information game



Minimax

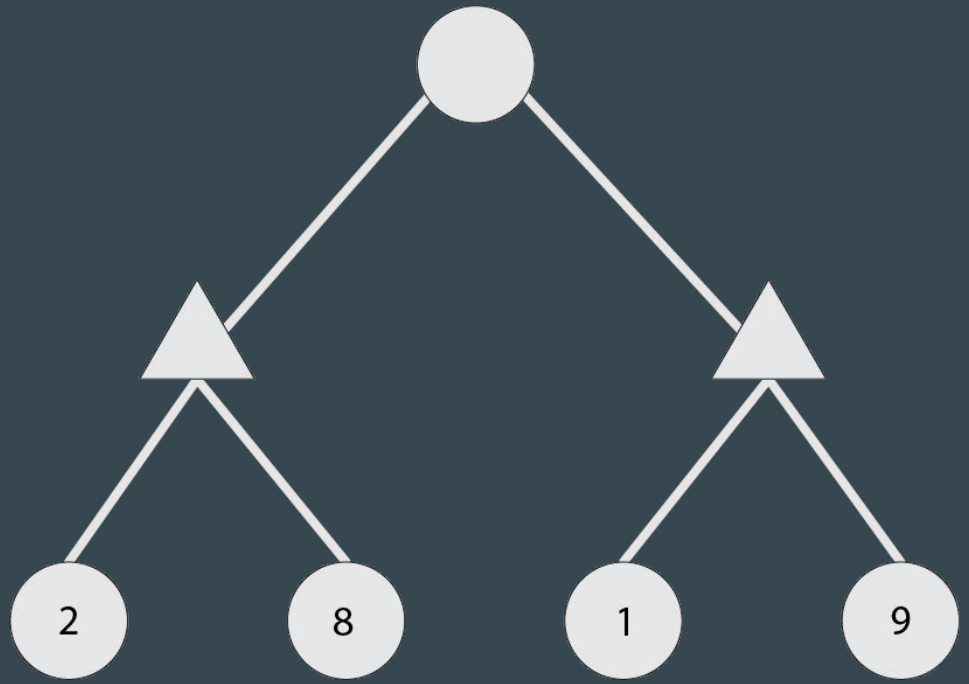
Minimax

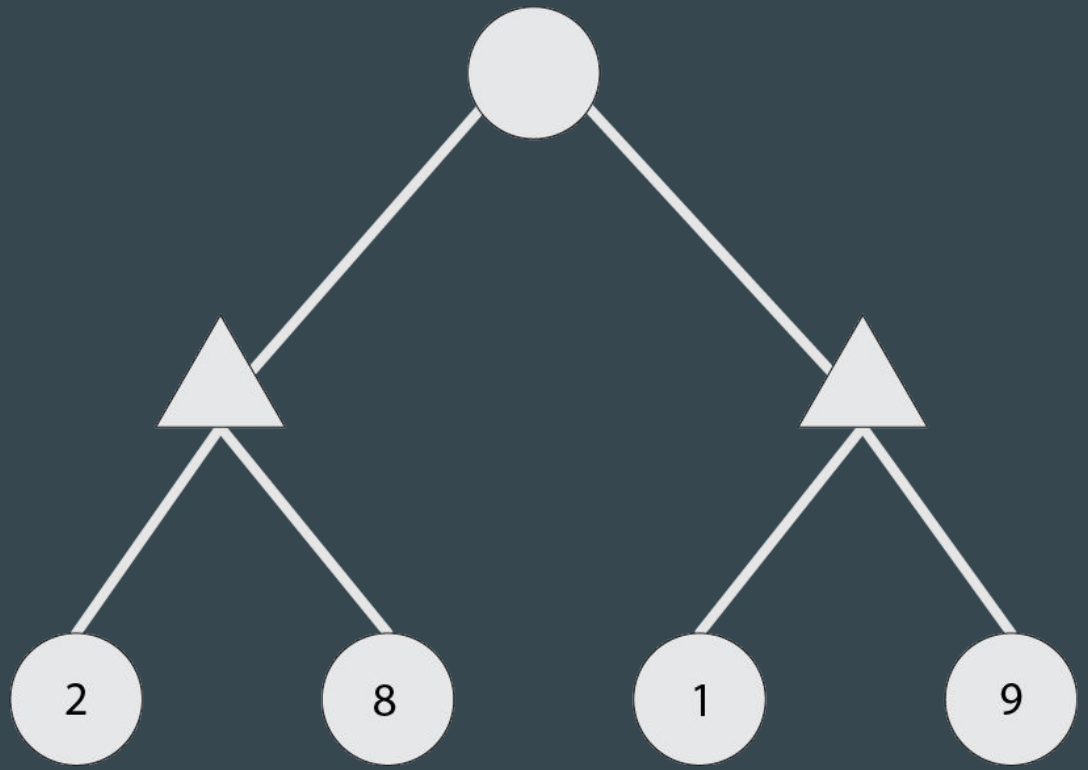
- The minimax algorithm is a decision rule used to minimize the loss for the worst possible case
- It returns the choice the highest guaranteed value without knowing the moves of the other players

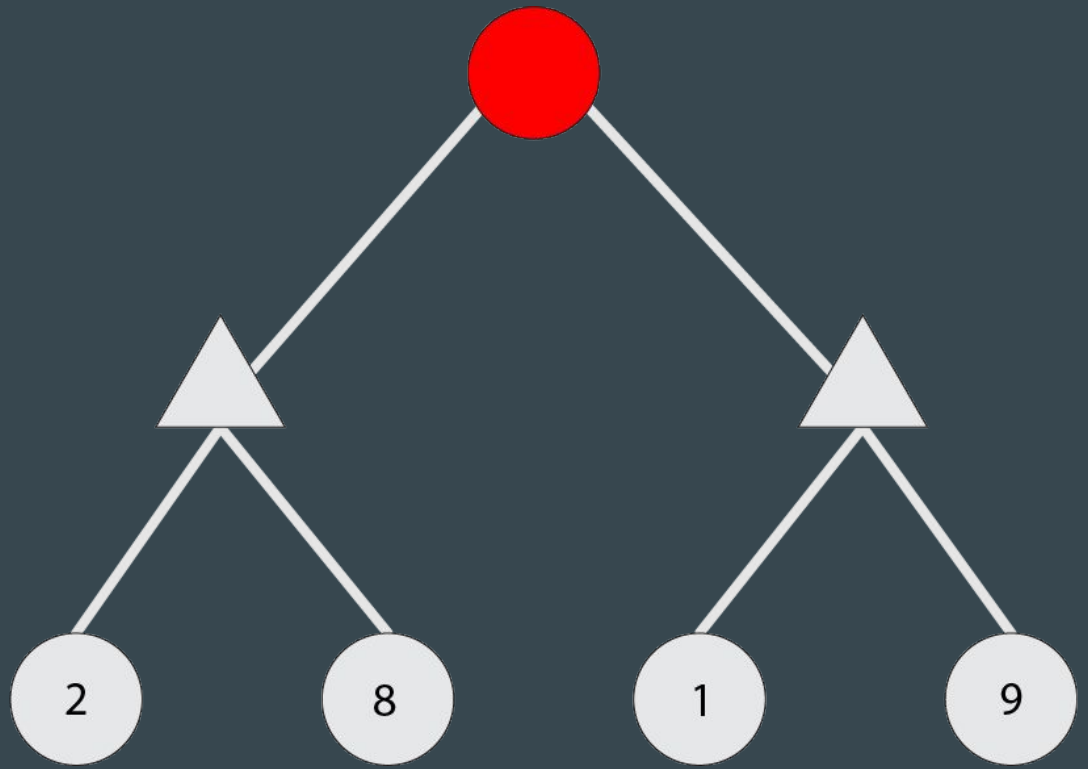


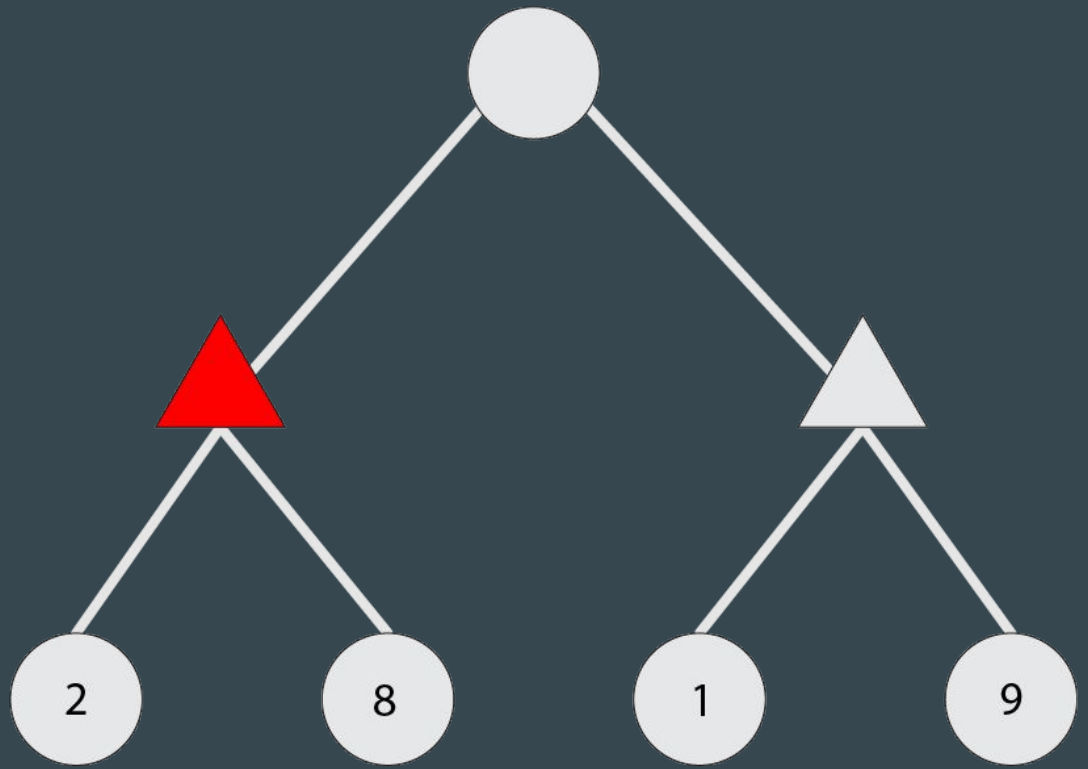
Minimax

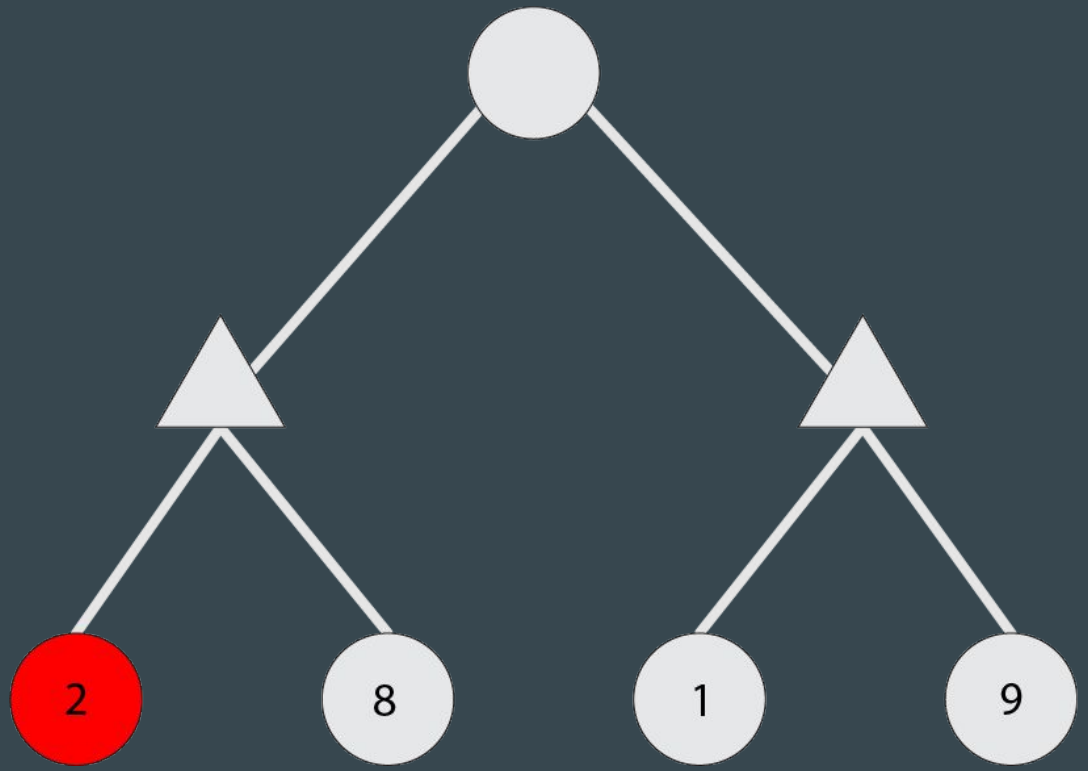
- Uses the adversarial search tree
- Space complexity of $O(b^m)$
- b = max branching factor of the search tree
- m = maximum depth of the search space
- Chess is a finite game

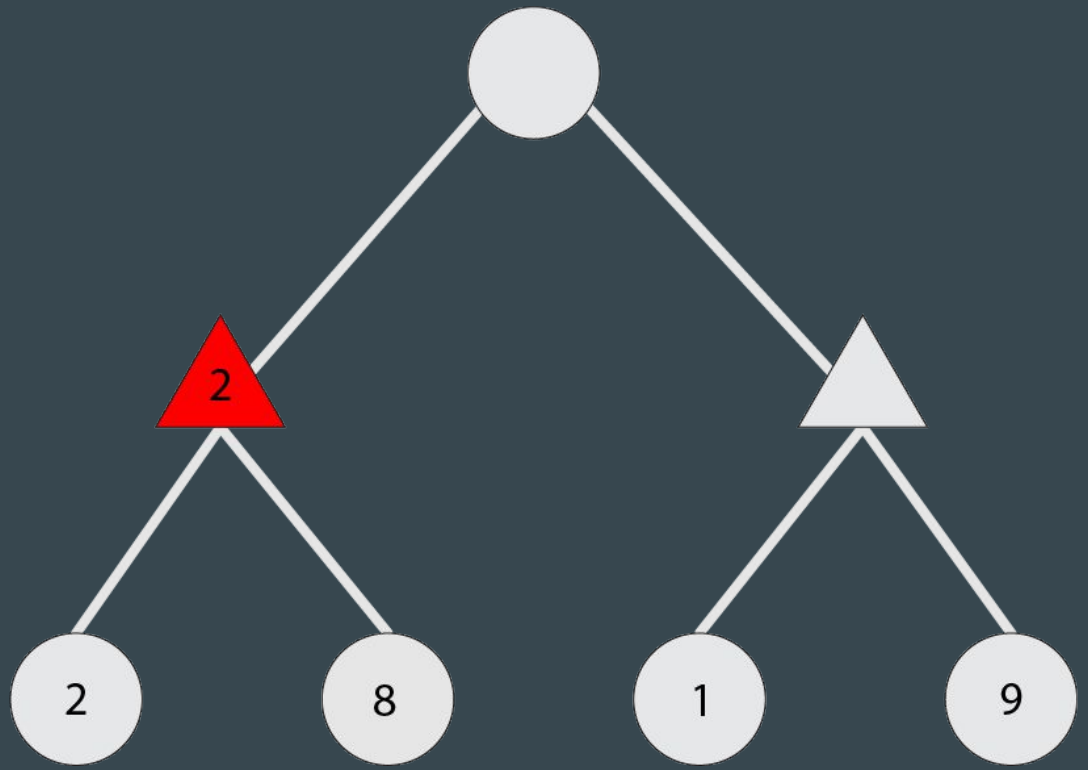


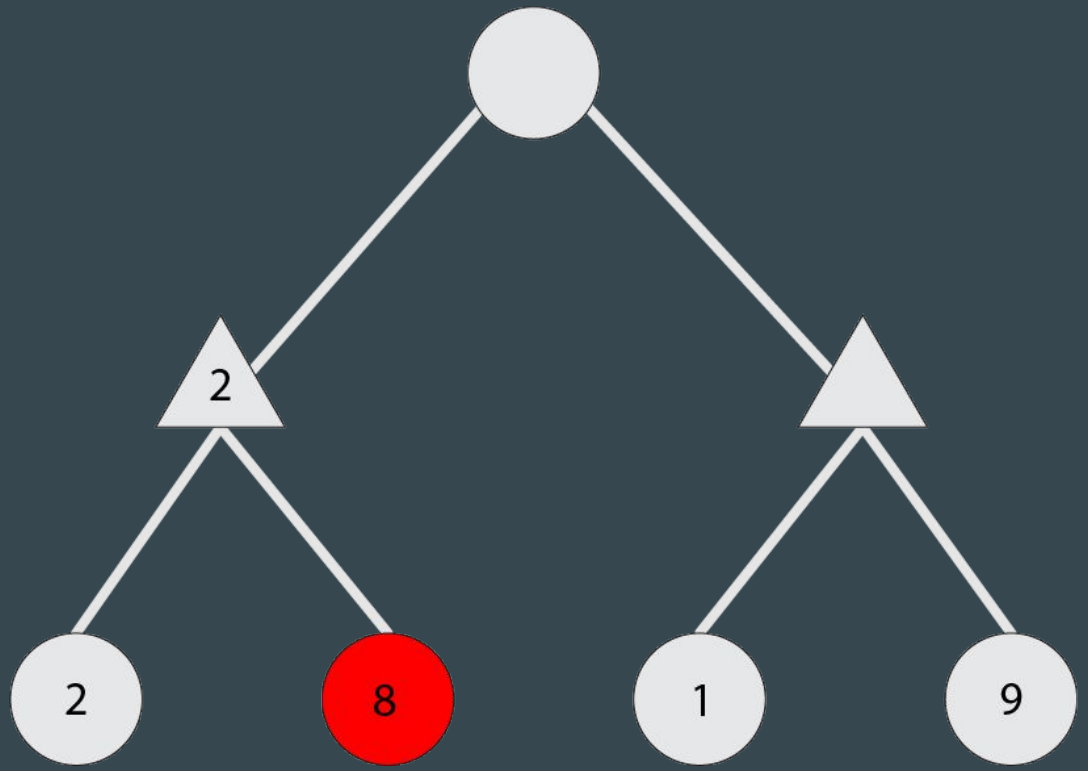

2
8
1
9

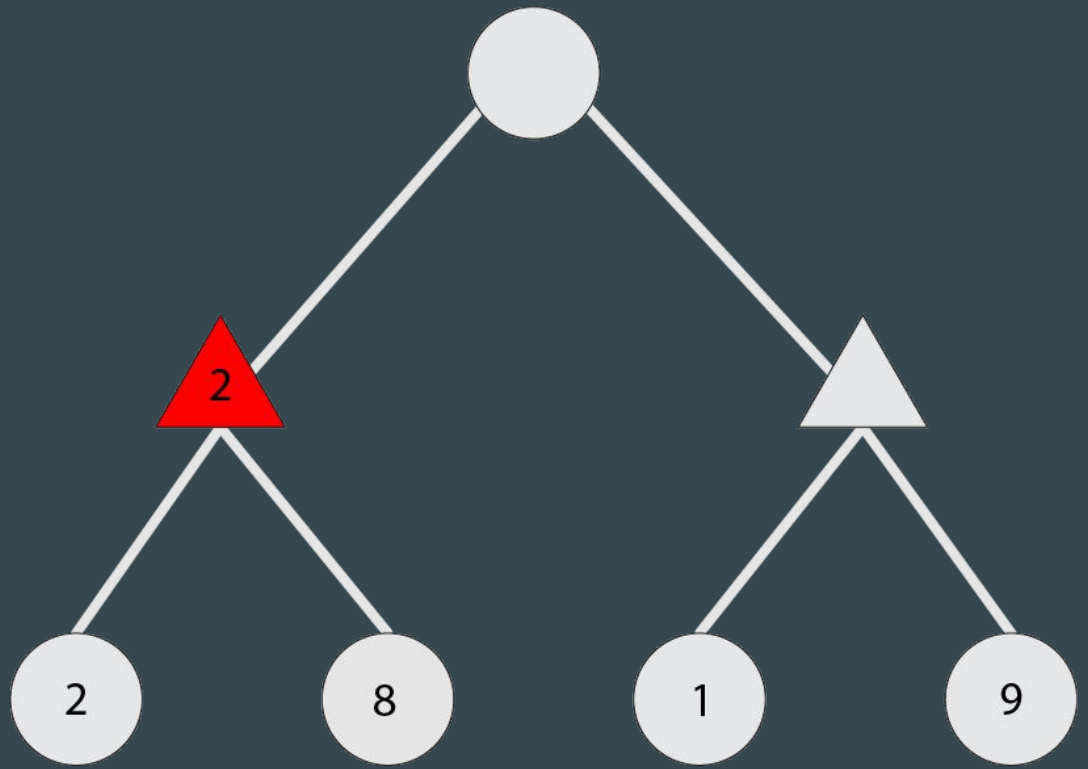

2
8
1
9

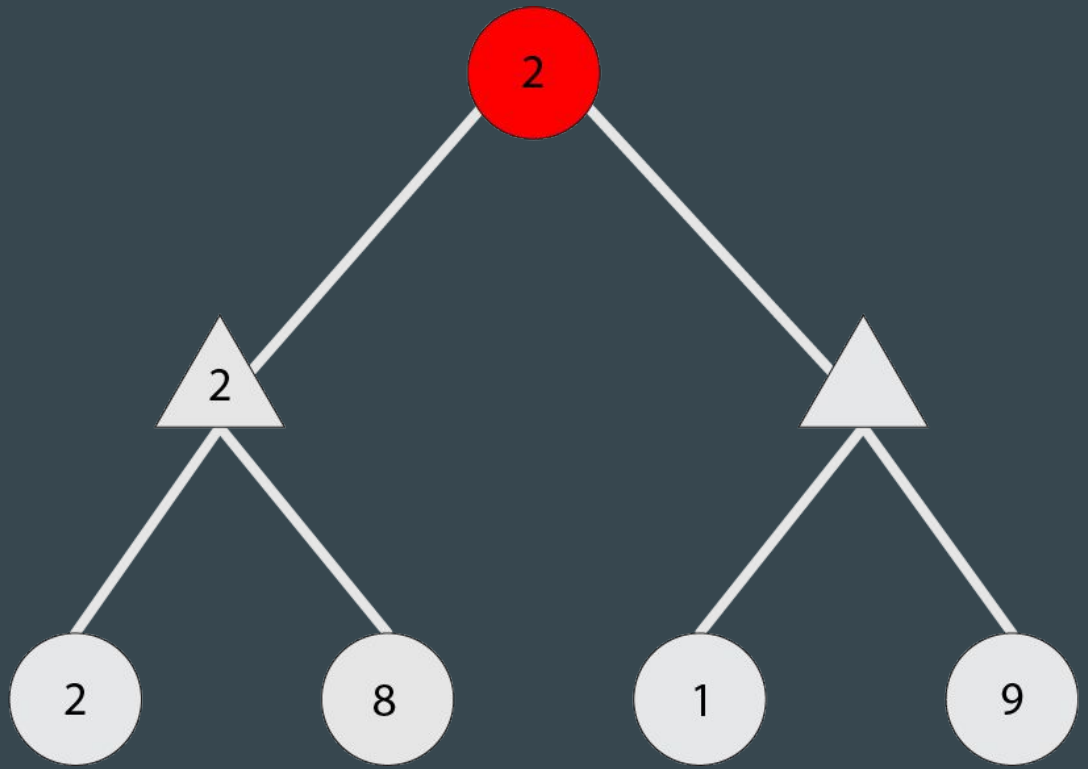

2
8
1
9

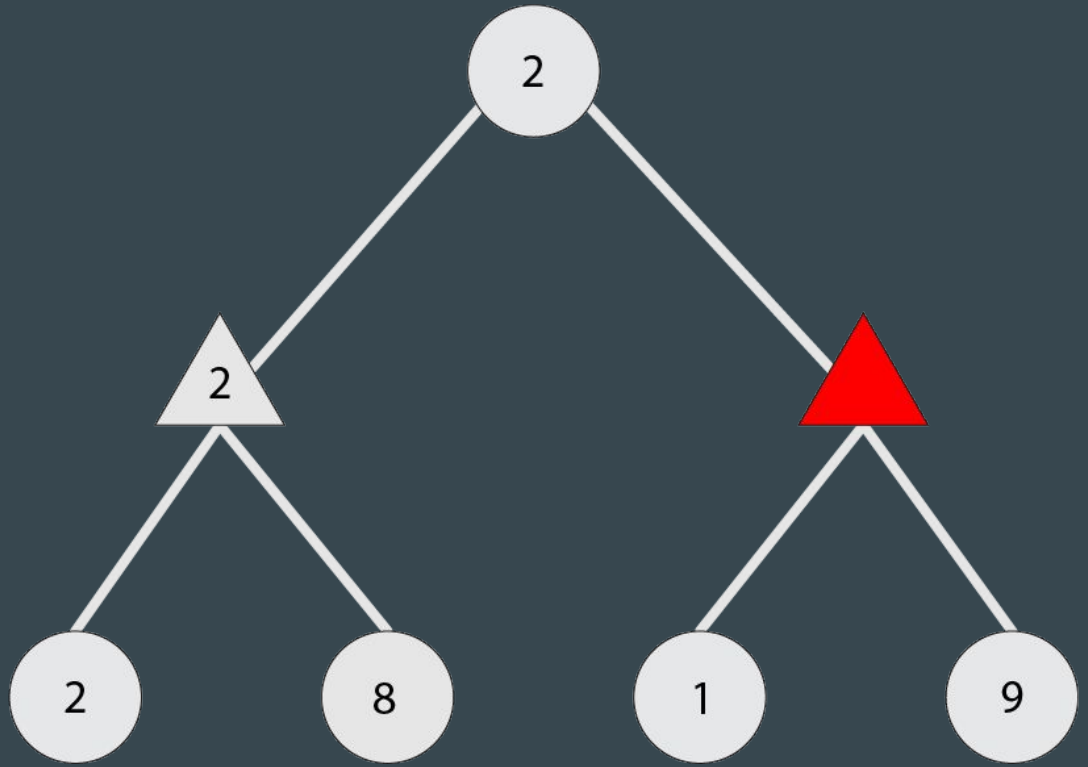

2
8
1
9

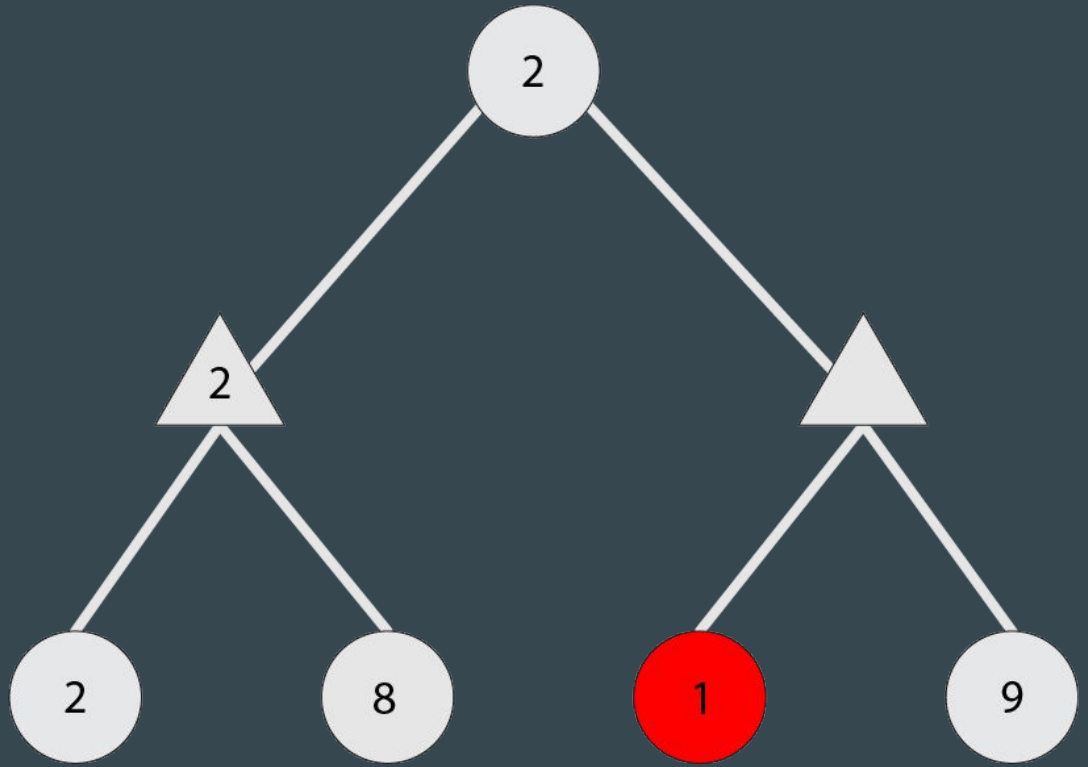

2
2
8
1
9

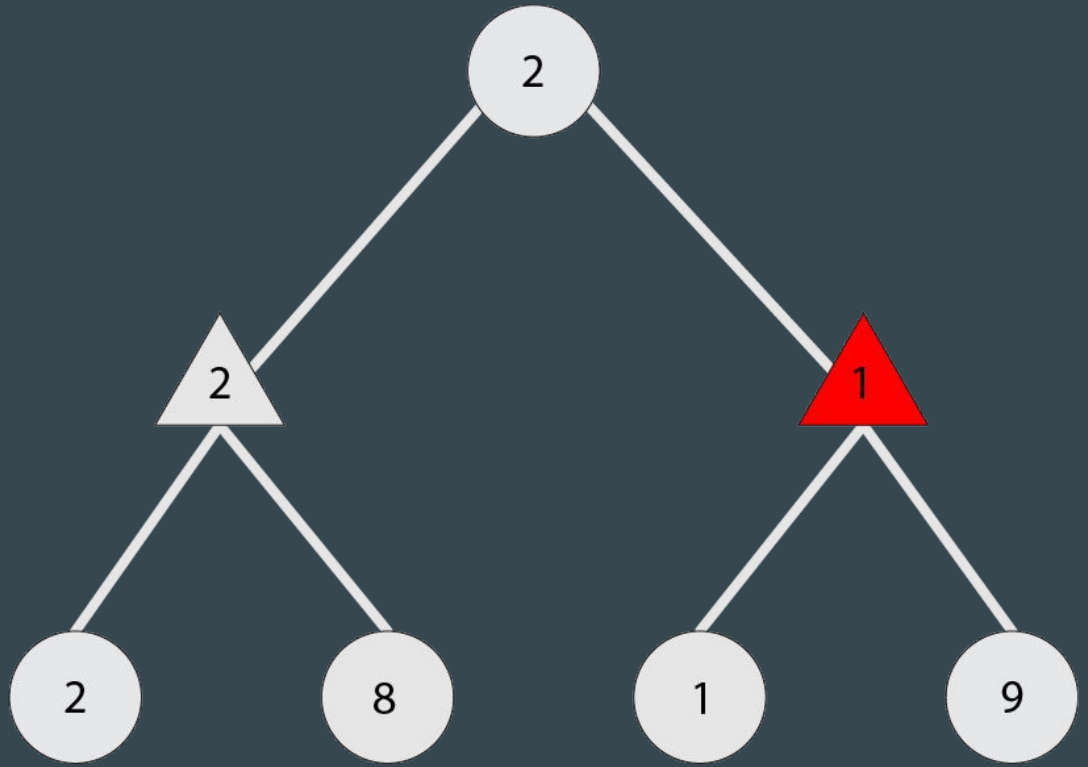

2
2
8
1
9

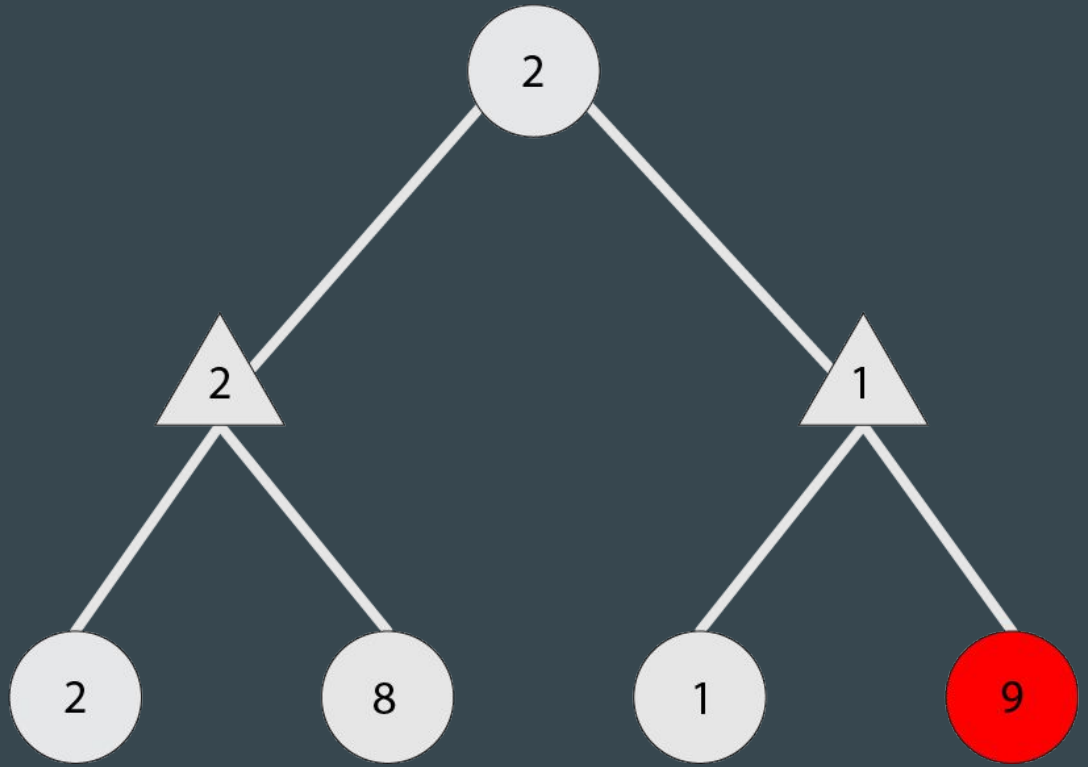

2
2
8
1
9

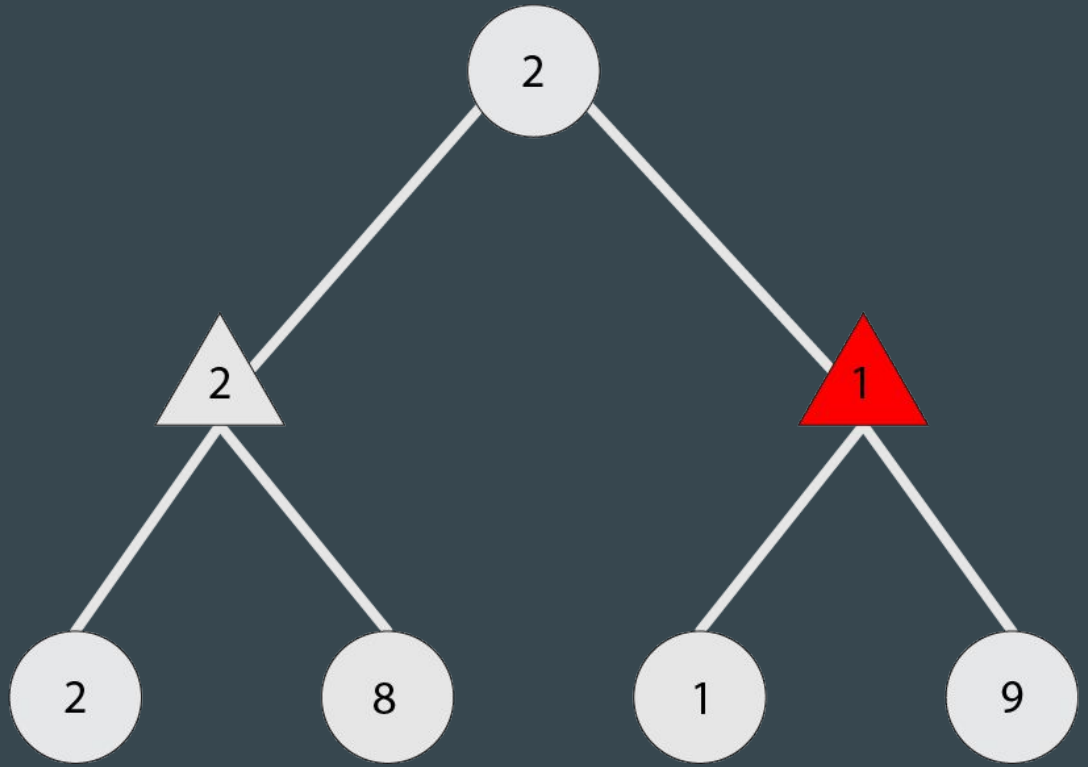

2
2
2
8
1
9

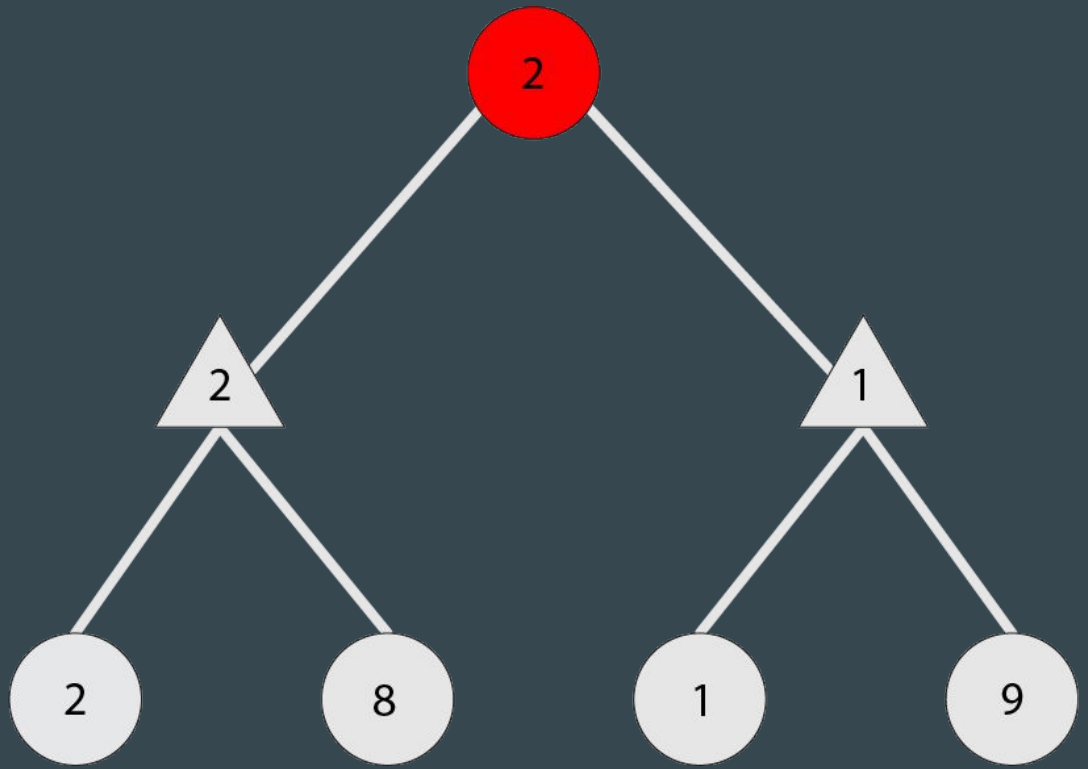

2
2
2
8
1
9

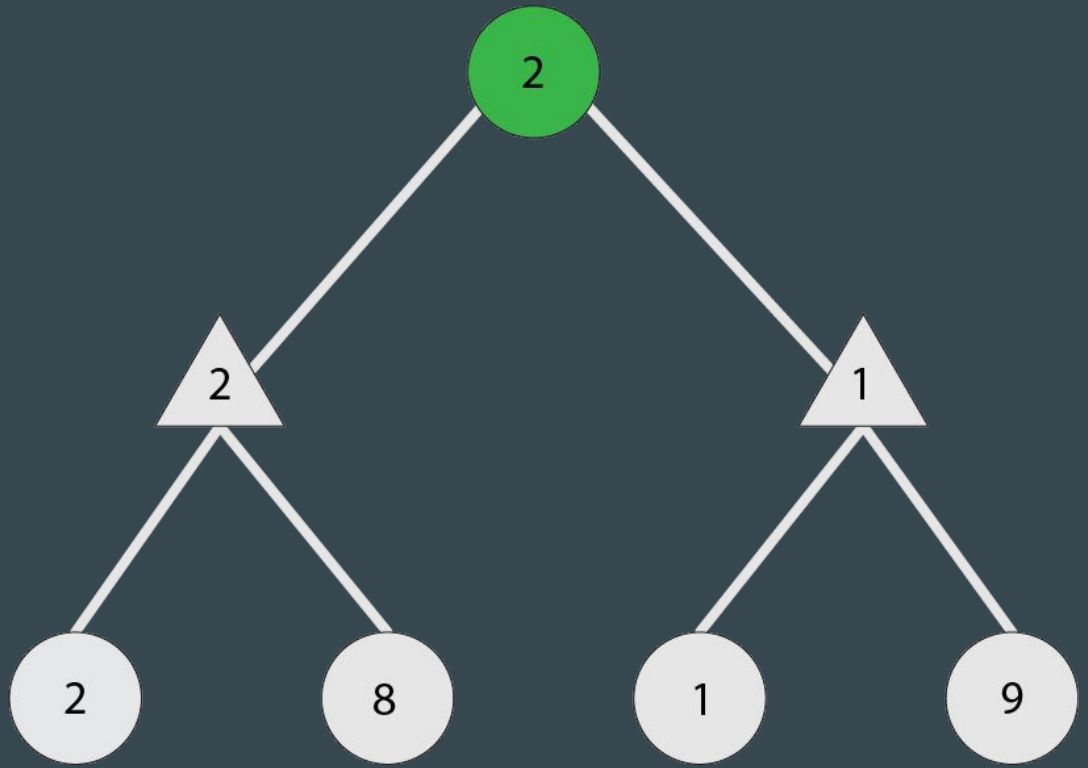

2
2
2
8
1
9


2
2
1
2
8
1
9


2
2
1
2
8
1
9


2
2
1
2
8
1
9

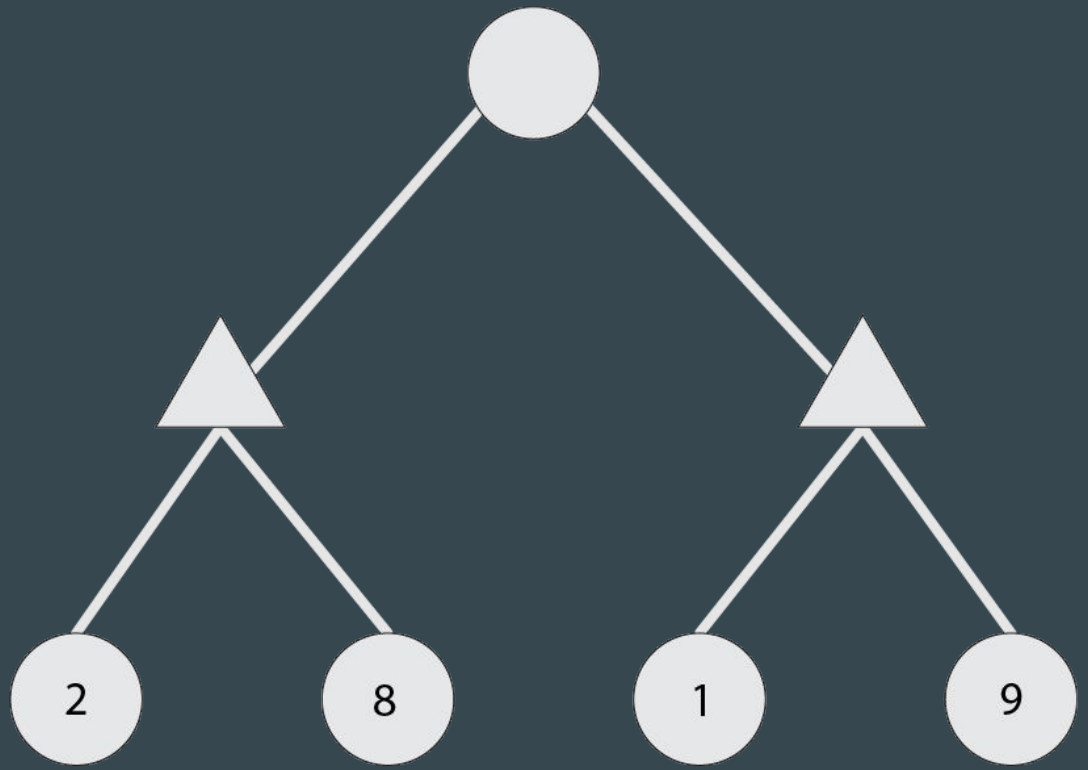

2
2
1
2
8
1
9


2
2
1
2
8
1
9

Alpha-Beta

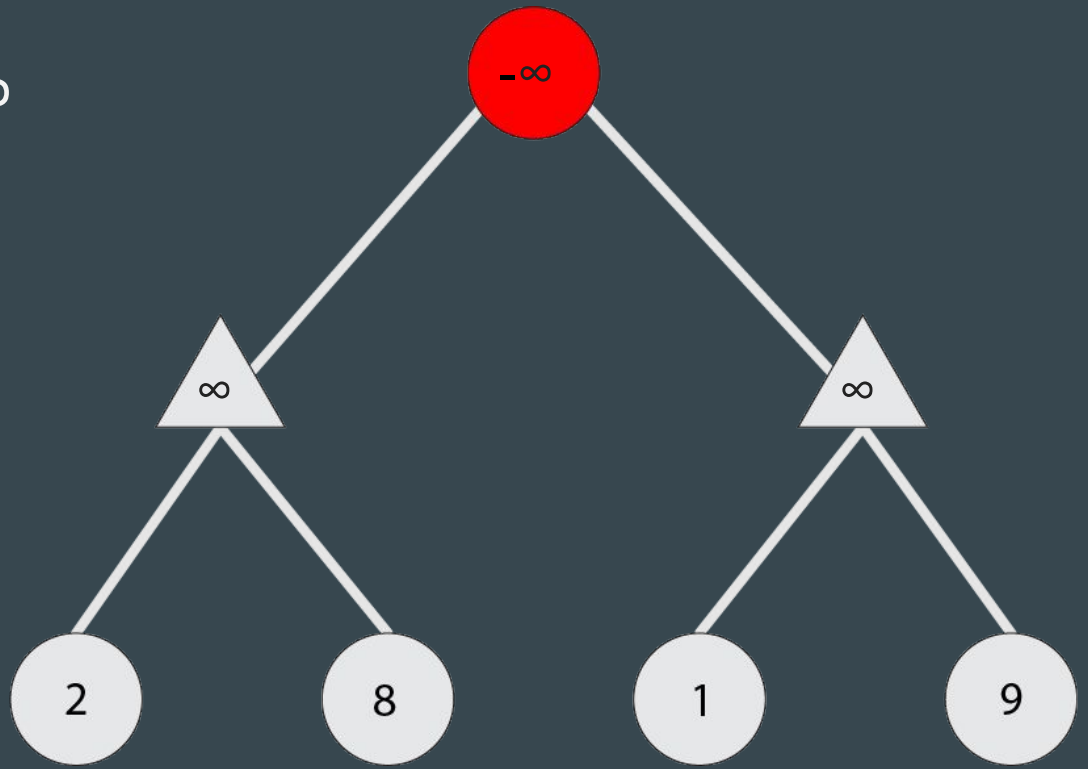
Alpha-Beta

- Alpha-Beta is an improvement of minimax
- It allows for faster pruning
- Space complexity of $O(b^{m/2})$ in best case, $O(b^{m})$ in worst case
- Alpha value: best value for current player found so far
- Beta value: best value for opponent found so far


2
8
1
9

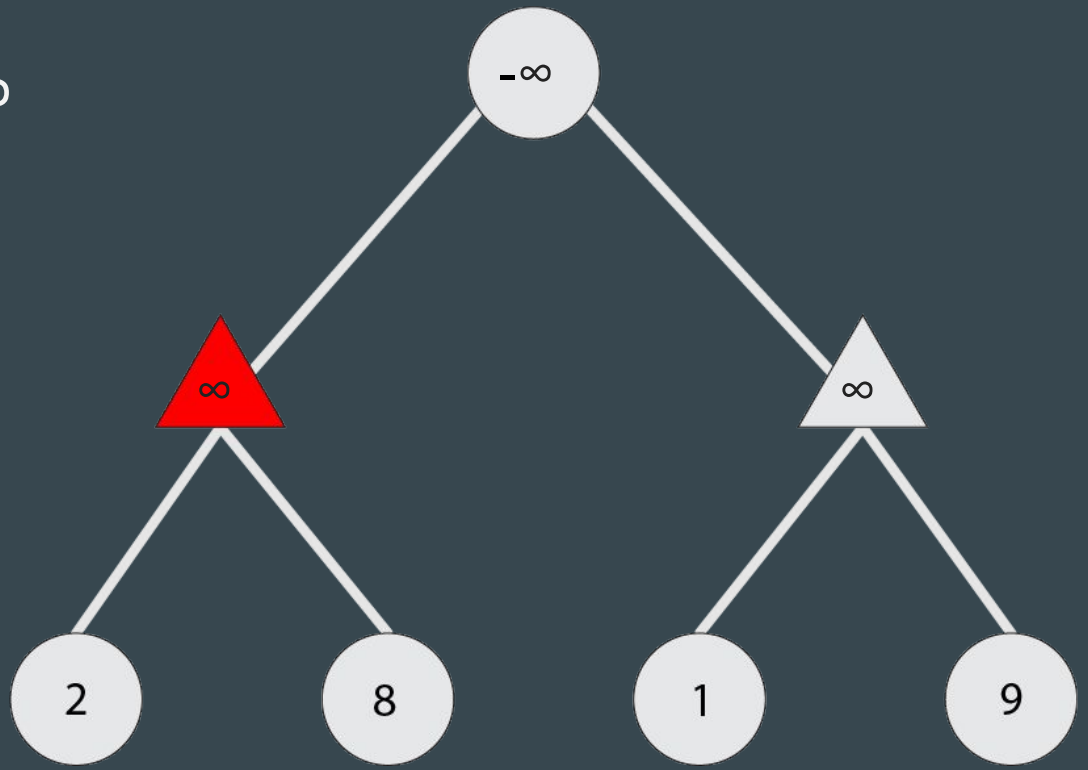
$\alpha = -\infty$

$\beta = \infty$



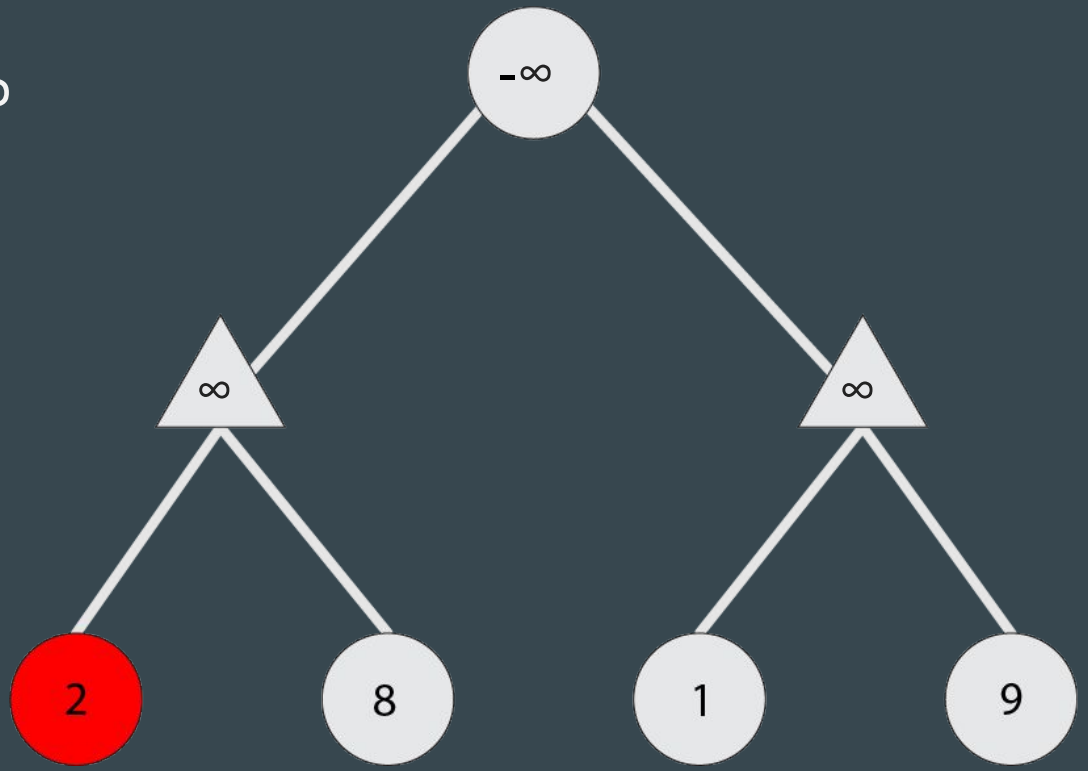
$\alpha = -\infty$

$\beta = \infty$



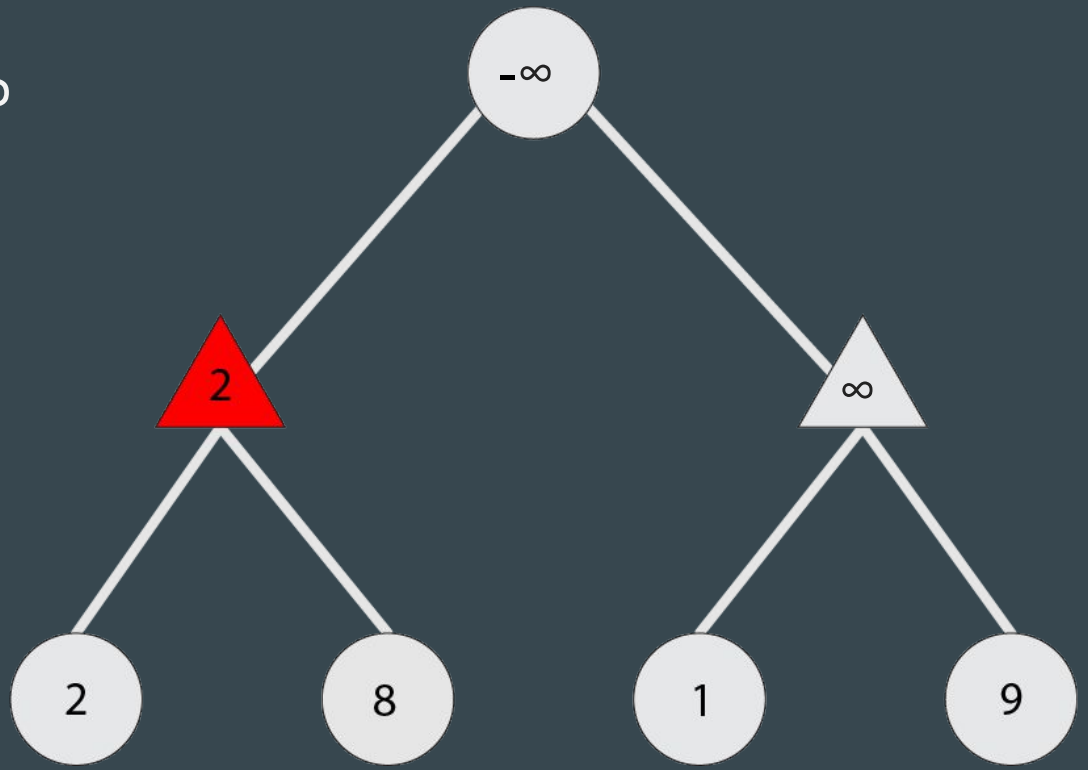
$\alpha = -\infty$

$\beta = \infty$



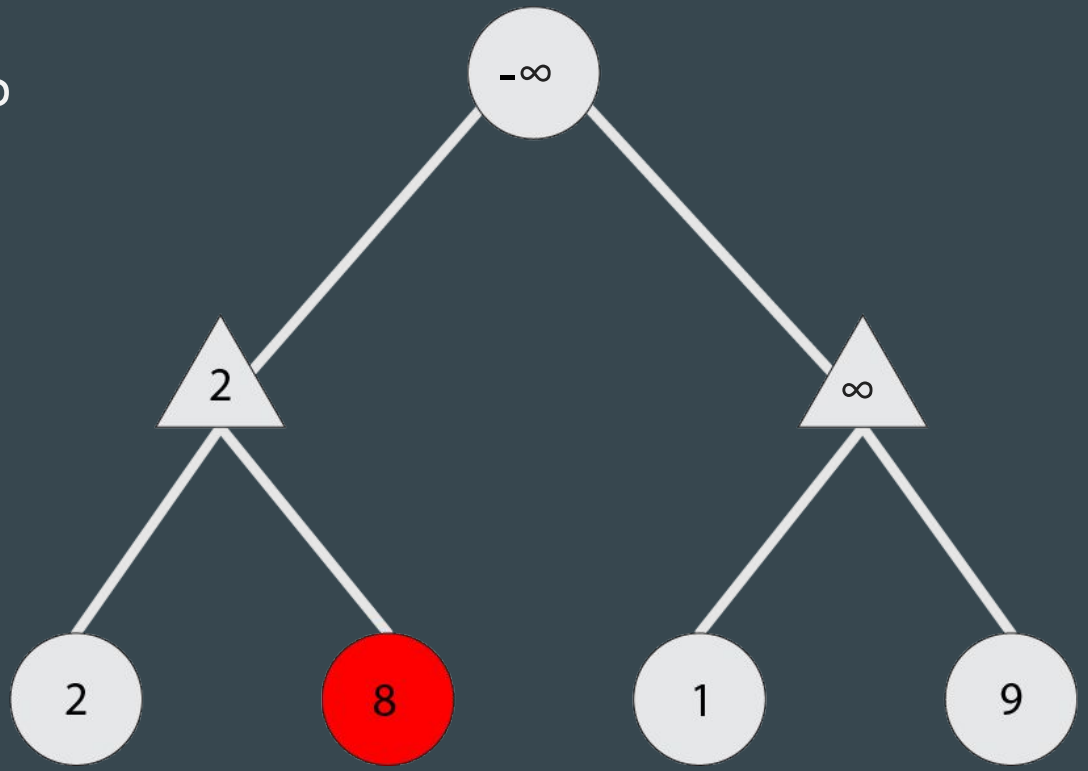
$\alpha = -\infty$

$\beta = 2$



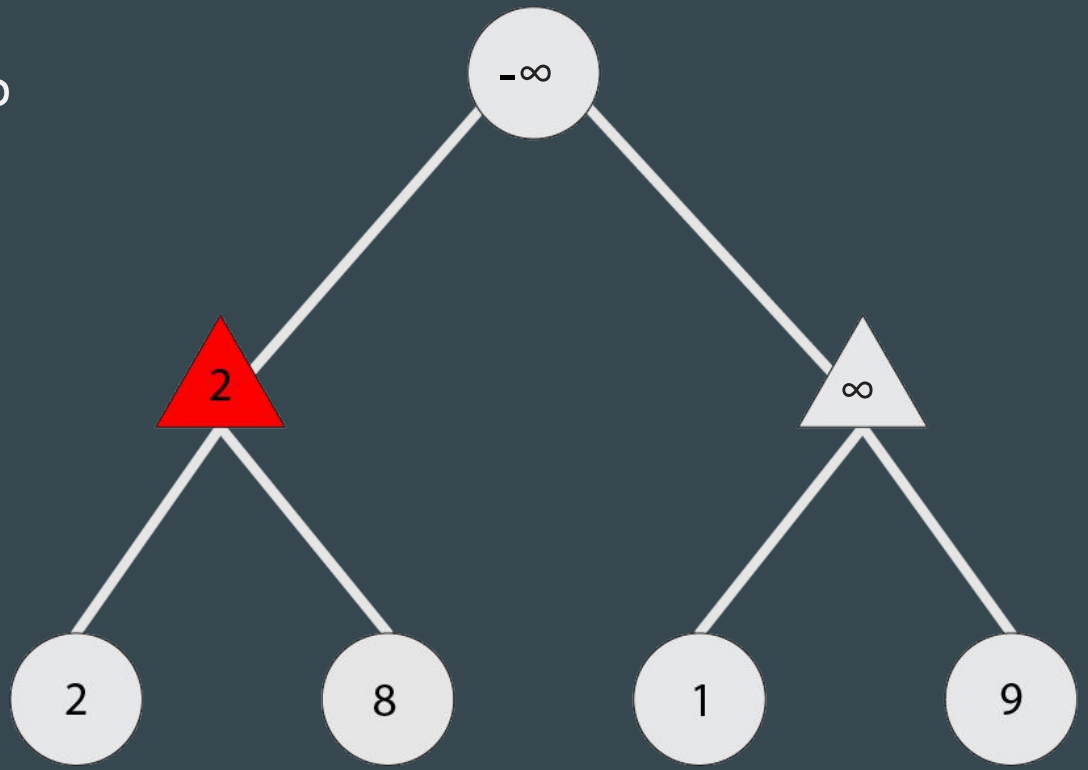
$\alpha = -\infty$

$\beta = 2$

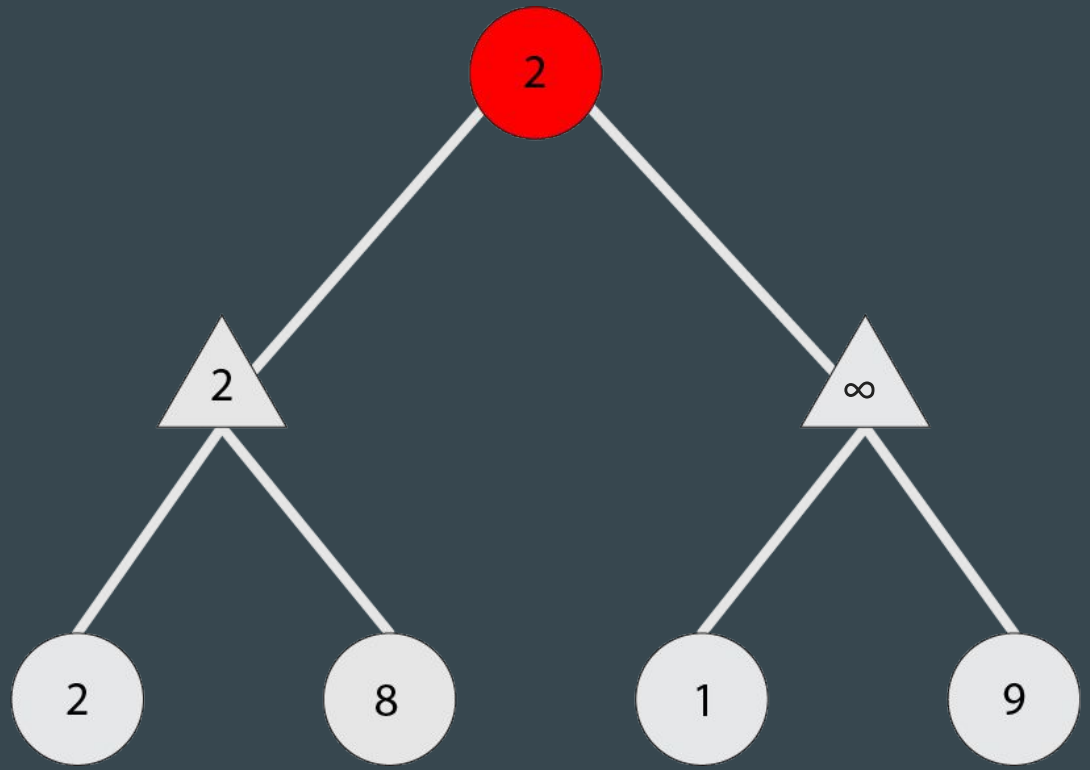


$\alpha = -\infty$

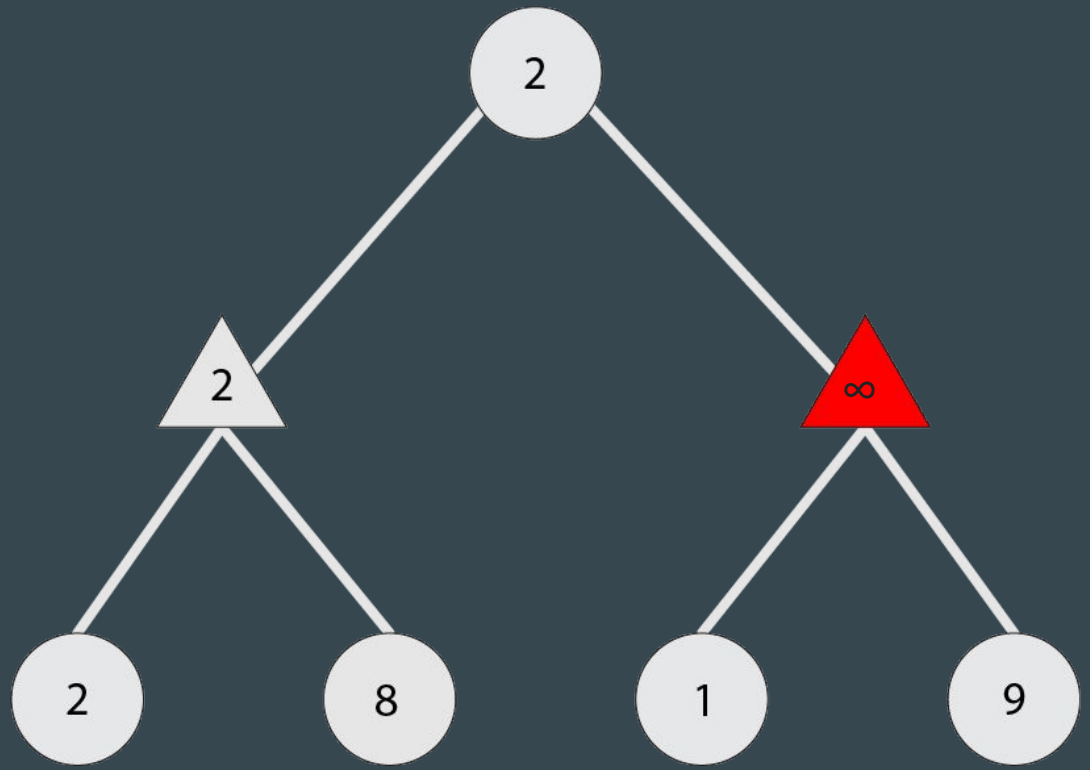
$\beta = 2$



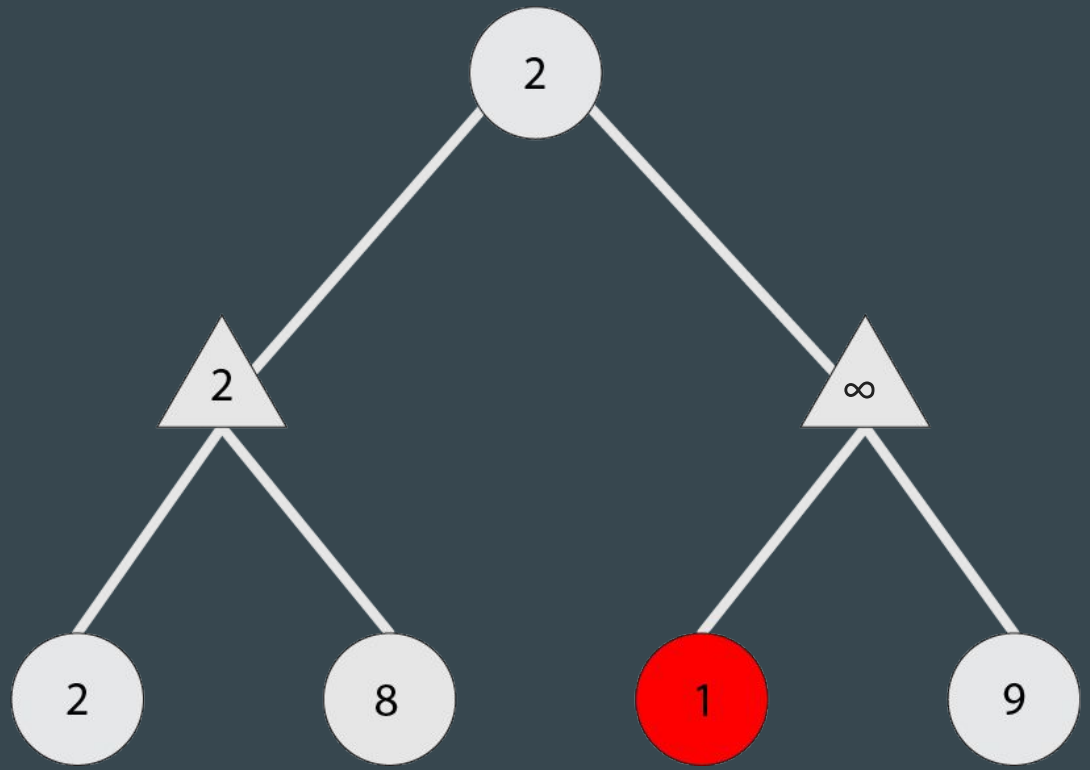
α = 2
β = 2



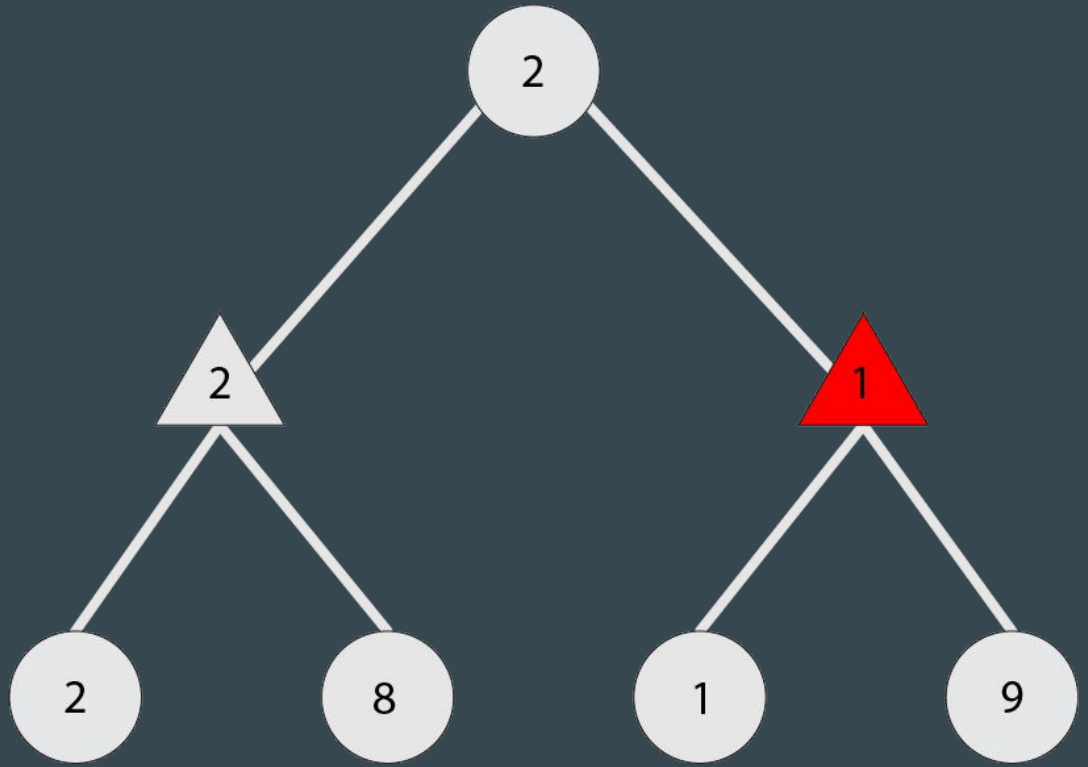
$\alpha = 2$
$\beta = \infty$



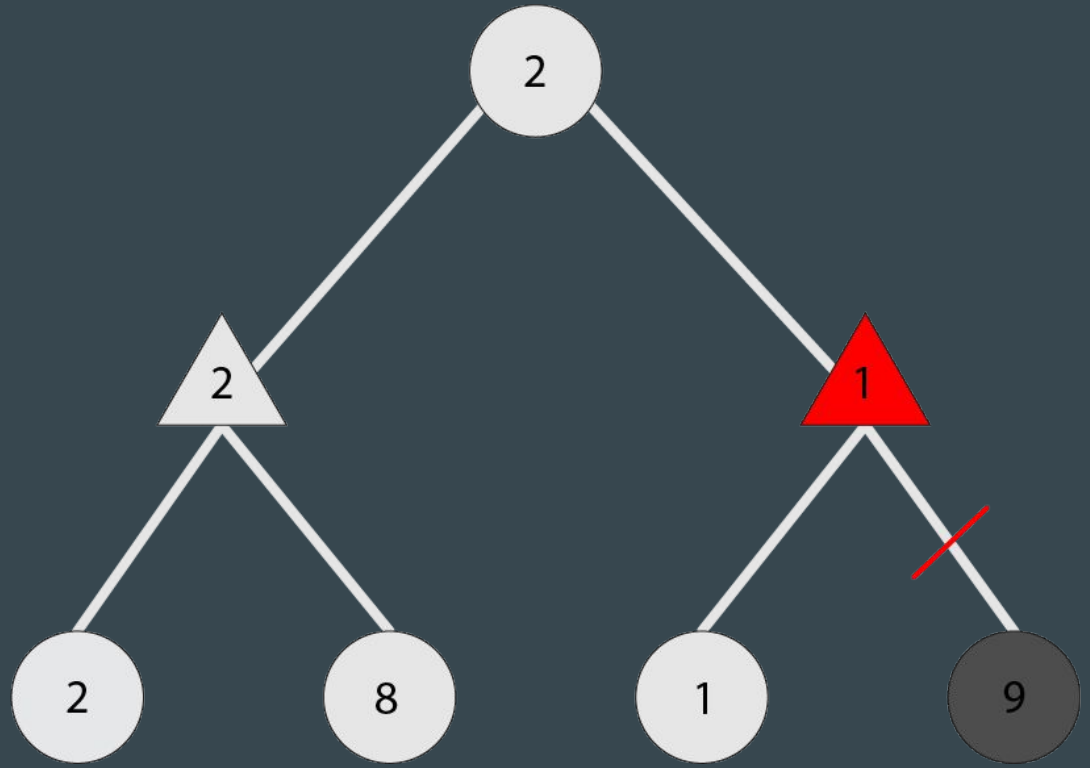
$\alpha = 2$
$\beta = \infty$



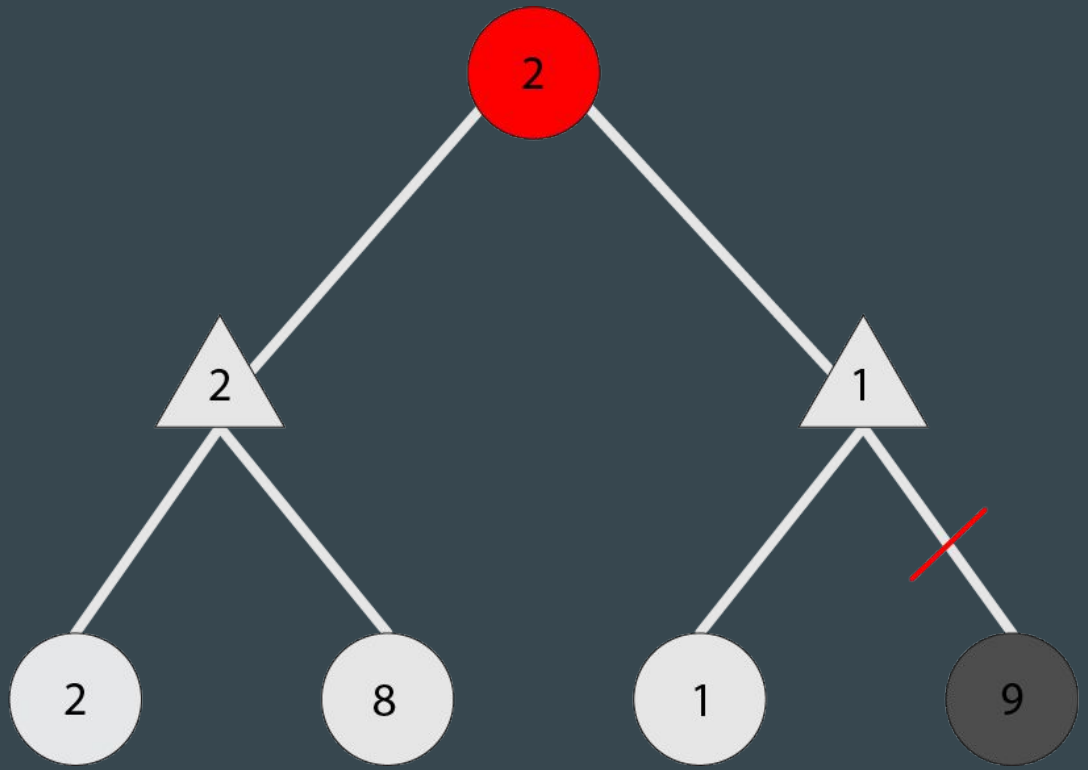
α = 2
β = 1



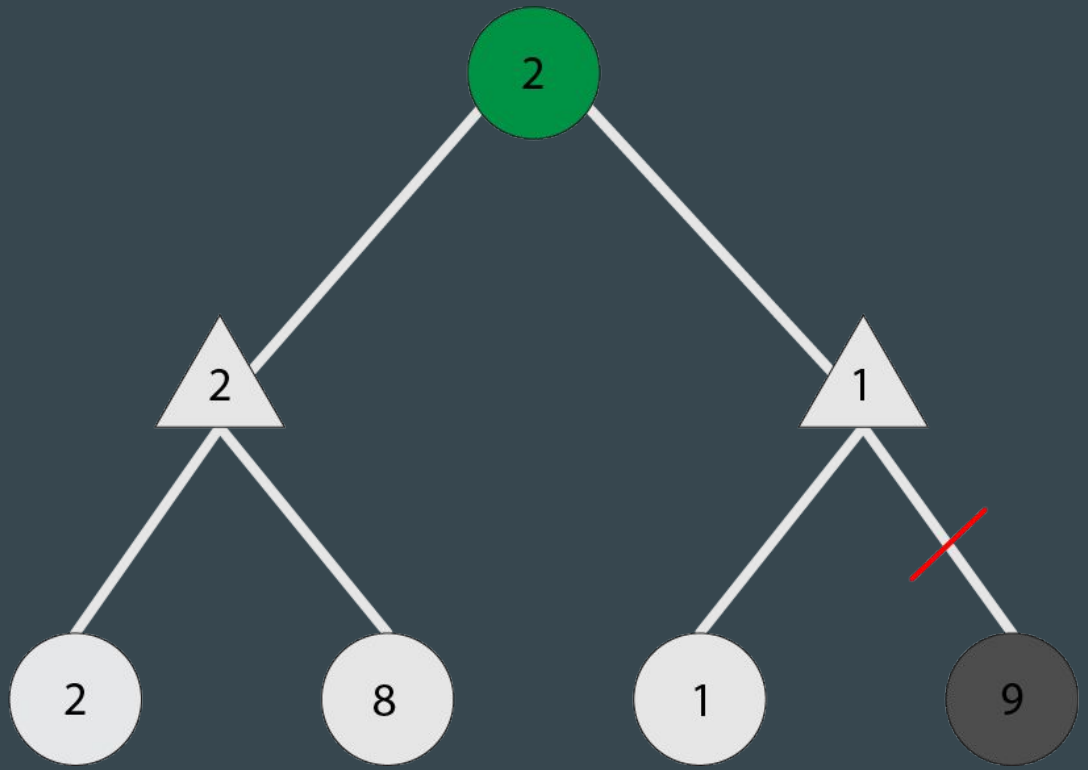
α = 2
β = 1

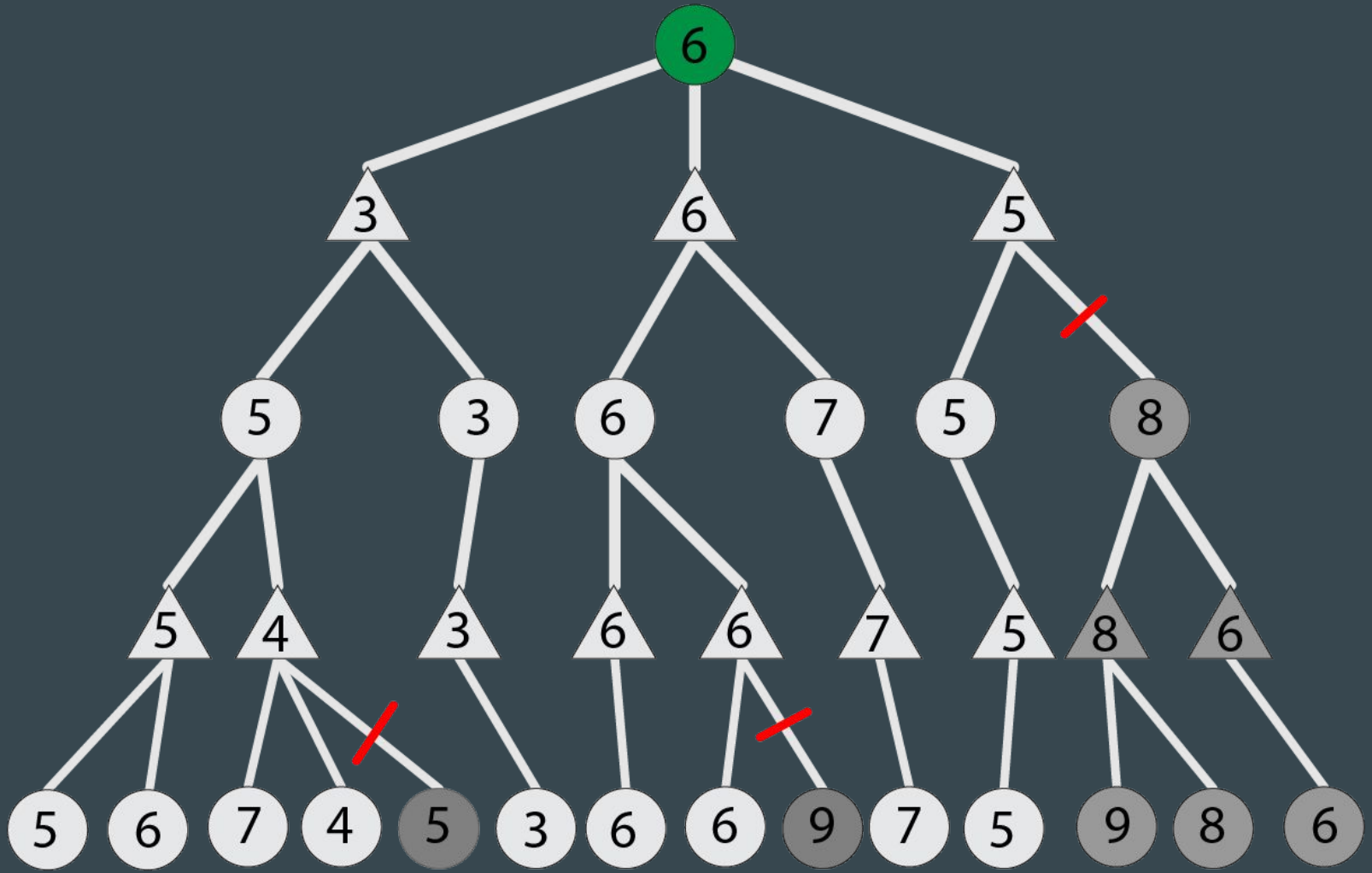


α = 2
β = 1



α = 2
β = 1




6
3
6
5
5
3
6
7
5
8
5
4
3
6
6
7
5
8
6
5
6
7
4
5
3
6
6
9
7
5
9
8
6

Implementations in Chess Engines

What do we need to make a chess engine

- Board representation
- Evaluating position
- Alpha-Beta
- Enhancements
- Chess shortcuts and improvements

Board Representation

Piece Centric

- Piece-Lists
- Piece-Sets
- Bitboards

Board Centric

- Mailbox
- 8x8 Board
- 10x12 Board
- 0x88
- Vector Attacks

Board Representation

Piece Centric

- Piece-Lists

  Lists of the pieces on the board

Board Representation

Piece Centric

- Piece-Lists
- Piece-Sets

  One bit for each piece inside a
  32-bit word, similar to bitboards

Board Representation

Piece Centric

- Piece-Lists
- Piece-Sets
- Bitboards

  64 bit sets with one for each piece

Board Representation

Piece Centric

- Piece-Lists
- Piece-Sets
- Bitboards

Board Centric

- Mailbox

  Every square in a separately addressable memory element

Board Representation

Piece Centric

- Piece-Lists
- Piece-Sets
- Bitboards

Board Centric

- Mailbox
- 8x8 Board

  A two-dimensional array containing piece and empty square codes

Board Representation

Piece Centric

- Piece-Lists
- Piece-Sets
- Bitboards

Board Centric

- Mailbox
- 8x8 Board
- 10x12 Board

  Similar to 8x8 with more squares added to allow for knight jumps as moves

Board Representation

Piece Centric

- Piece-Lists
- Piece-Sets
- Bitboards

Board Centric

- Mailbox
- 8x8 Board
- 10x12 Board
- 0x88

  This uses one nibble to index the piece or empty square

Board Representation

Piece Centric

- Piece-Lists
- Piece-Sets
- Bitboards

Board Centric

- Mailbox
- 8x8 Board
- 10x12 Board
- 0x88

Evaluating Position

Historically uses an evaluation polynomial:

$$V = c_1 f_1 + c_2 f_2 + \ldots + c_n f_n$$

Recent advancements:

- Using Neural Networks for Evaluation *K Greer 2013*
- Better storage of board

Alpha-Beta

- This is the best search method found so far *K Greer 2013*
- How deep to go:
  - Deep blue beat Kasparov with a search depth of 14-16
  - Computers would need a search depth of 19 to consistently beat the current world champion
  - Modern computers search up to depth 35
- Move ordering, searching promising branches first

The impact of search depth on chess playing strength

Diogo R. Ferreira ICGA 2013

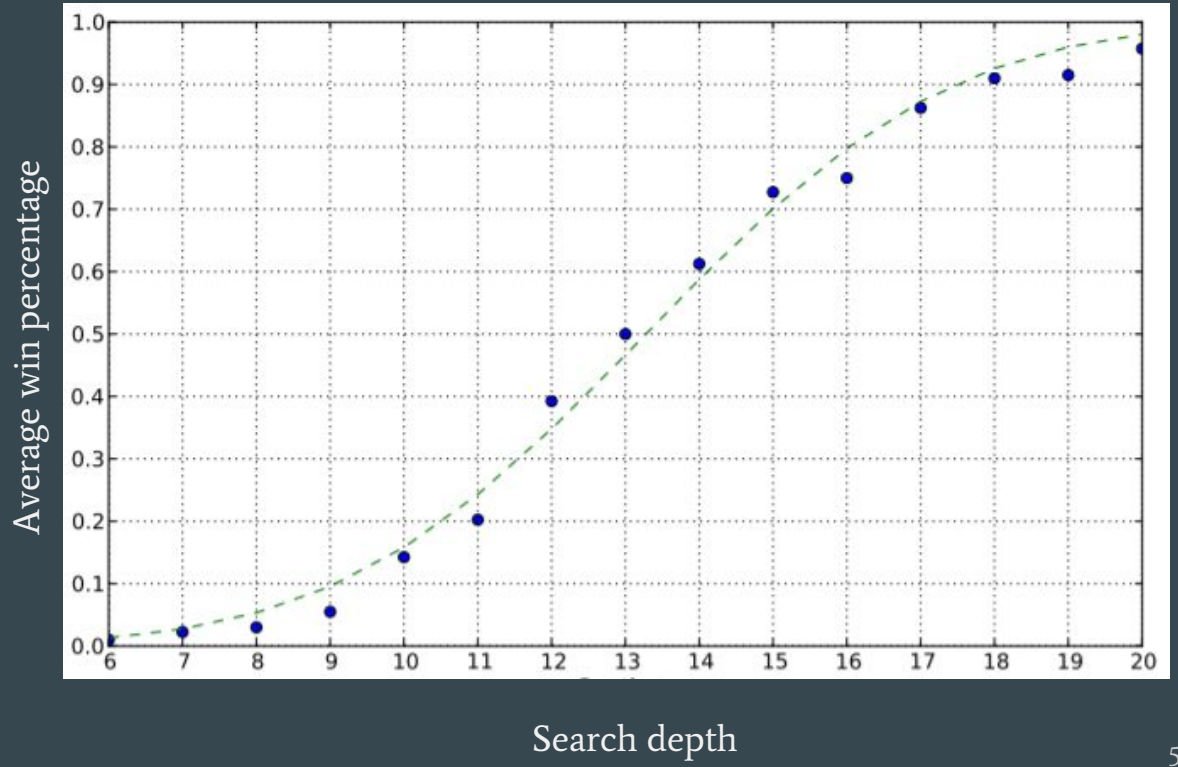
depth	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
6	0.50	0.29	0.15	0.07	0.02	0.03	0.01	0.01	0.00	0.01	0.00	0.01	0.00	0.00	0.00
7	0.71	0.50	0.30	0.16	0.07	0.04	0.01	0.02	0.01	0.00	0.00	0.00	0.01	0.00	0.00
8	0.84	0.70	0.50	0.24	0.12	0.06	0.04	0.03	0.01	0.01	0.01	0.00	0.01	0.01	0.01
9	0.93	0.84	0.76	0.50	0.28	0.18	0.10	0.06	0.04	0.03	0.01	0.01	0.01	0.00	0.00
10	0.98	0.93	0.89	0.72	0.50	0.32	0.26	0.14	0.08	0.06	0.03	0.03	0.01	0.01	0.01
11	0.97	0.96	0.94	0.81	0.68	0.50	0.34	0.20	0.15	0.11	0.09	0.06	0.03	0.03	0.03
12	0.99	0.98	0.96	0.90	0.74	0.66	0.50	0.39	0.27	0.19	0.12	0.08	0.04	0.04	0.04
13	0.99	0.98	0.97	0.94	0.86	0.80	0.61	0.50	0.39	0.27	0.25	0.14	0.09	0.09	0.04
14	1.00	0.99	0.99	0.95	0.92	0.85	0.73	0.61	0.50	0.38	0.30	0.21	0.16	0.15	0.07
15	0.99	1.00	0.99	0.97	0.94	0.89	0.81	0.73	0.62	0.50	0.40	0.33	0.24	0.21	0.15
16	1.00	1.00	0.99	0.98	0.97	0.91	0.89	0.75	0.70	0.60	0.50	0.40	0.31	0.27	0.15
17	0.99	1.00	1.00	0.99	0.97	0.94	0.92	0.86	0.79	0.67	0.60	0.50	0.44	0.32	0.26
18	1.00	0.99	0.99	0.99	0.99	0.97	0.95	0.91	0.84	0.76	0.69	0.56	0.50	0.45	0.34
19	1.00	1.00	0.99	1.00	0.99	0.97	0.96	0.92	0.85	0.79	0.73	0.68	0.55	0.50	0.43
20	1.00	1.00	0.99	1.00	0.99	0.97	0.96	0.96	0.93	0.85	0.85	0.74	0.66	0.57	0.50

Average score over 200 games irrespective of color

The impact of search depth on chess playing strength

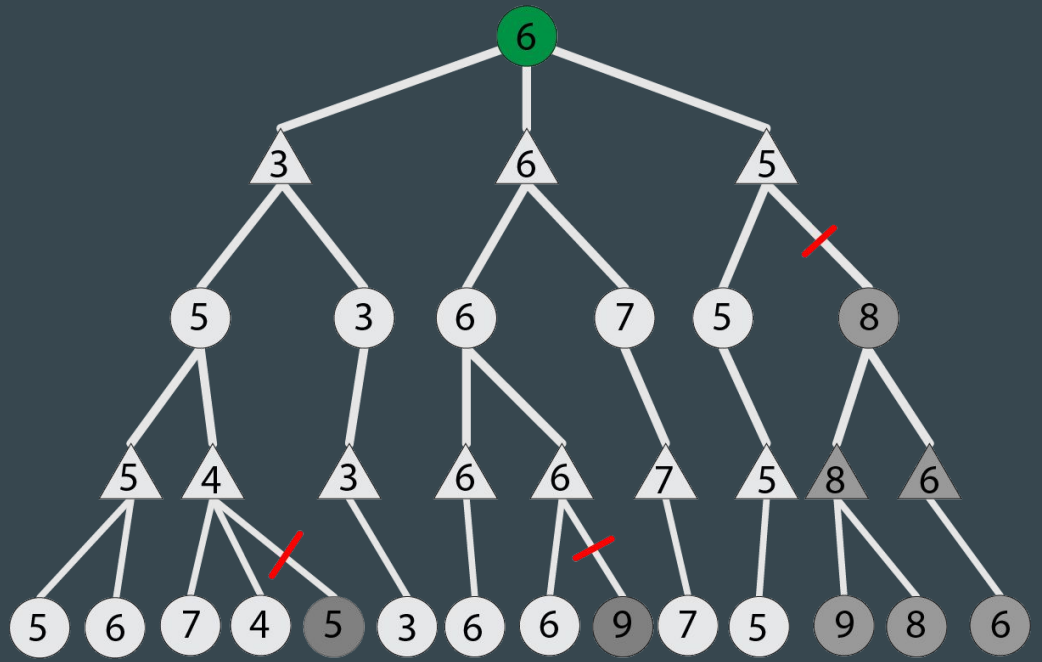
Diogo R. Ferreira ICGA 2013

- Each move added or removed significantly changes engine strength
- Played using HOUDINI 1.5a 64-bit
- A search depth of 20 is rated 2893.9, top human is 2838



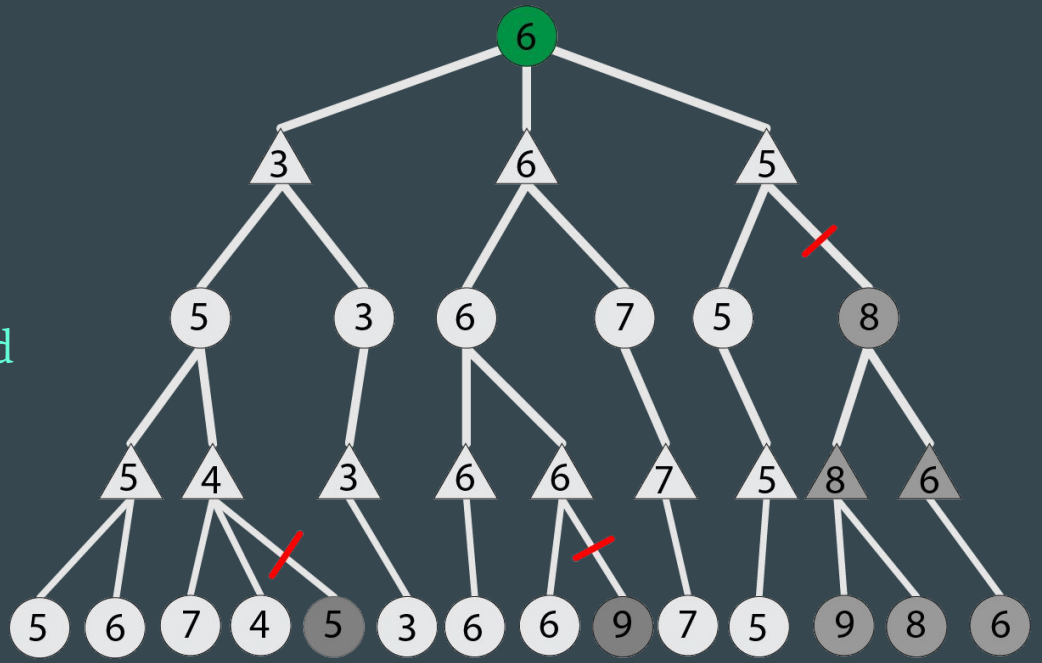
Enhancements

- Limiting search space
  - Starting with more promising branches and going deeper in these



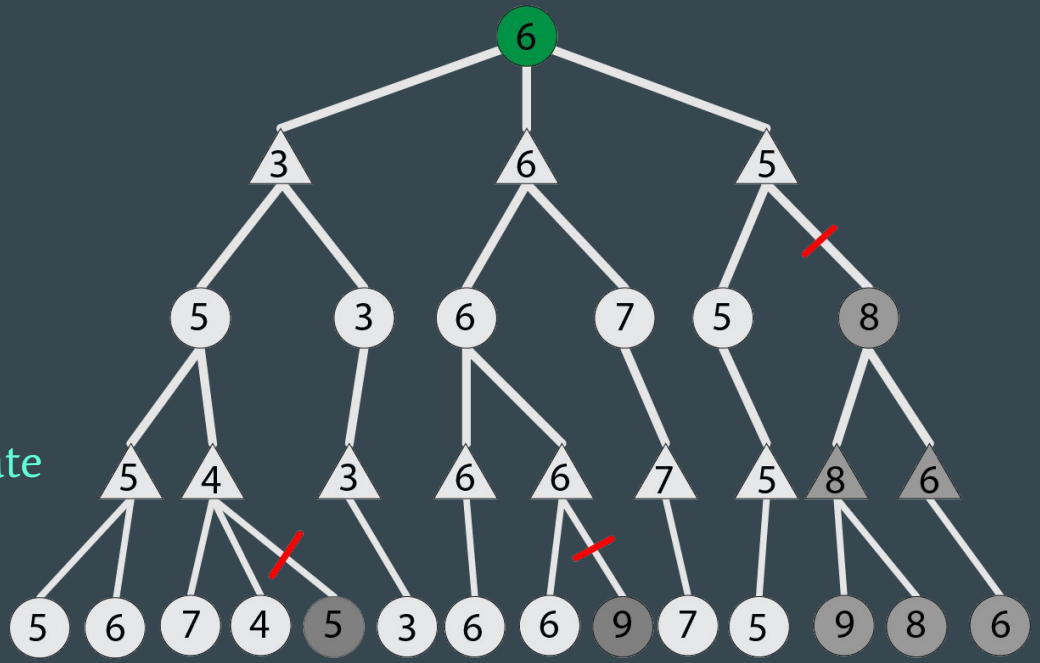
Enhancements

- Limiting search space
- Iterative deepening
  - Starting the search shallow and deepening



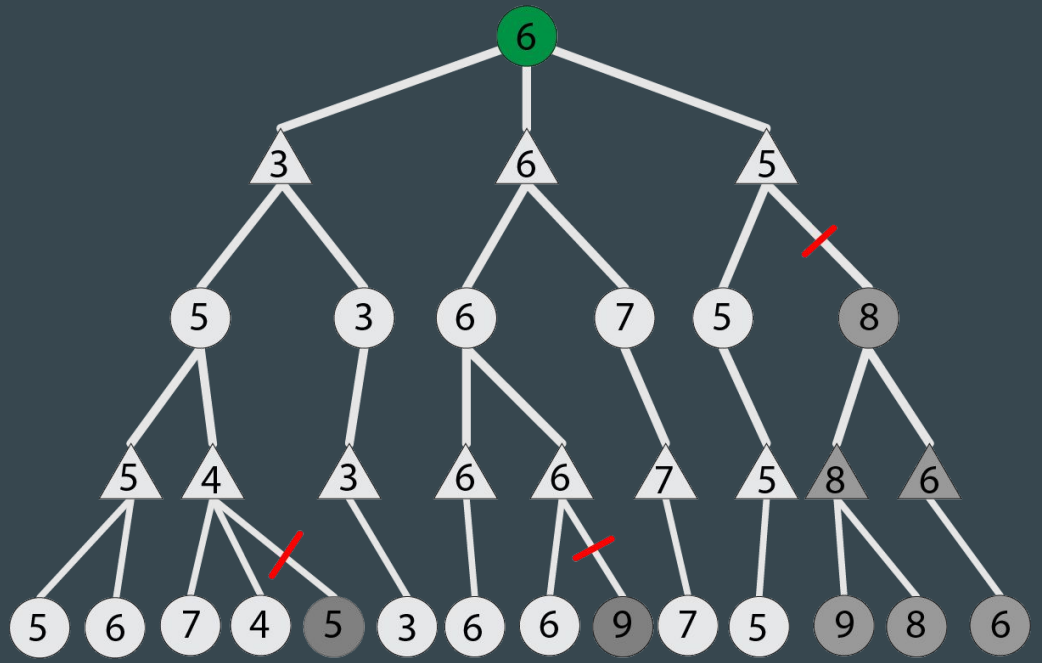
Enhancements

- Limiting search space
- Iterative deepening
- Parallel computing
  - Giving each branch to a separate core to speed up the search



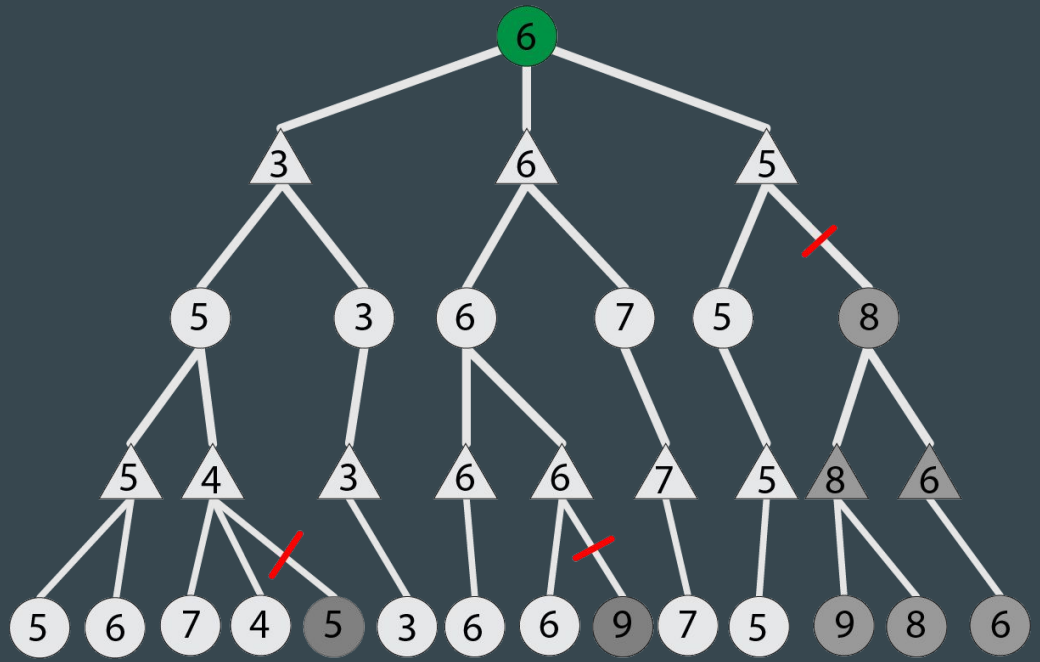
Enhancements

- Limiting search space
- Iterative deepening
- Parallel computing
- Uneven tree development
  - Chess is not the same depth every option



Enhancements

- Limiting search space
- Iterative deepening
- Parallel computing
- Uneven tree development



Chess Shortcuts and Improvements

- 3 parts of the game:
  - Opening
  - Middlegame
  - Endgame
- Principal Variation
  - A sequence of moves that programs consider best and expect to be played

Opening

Middle Game

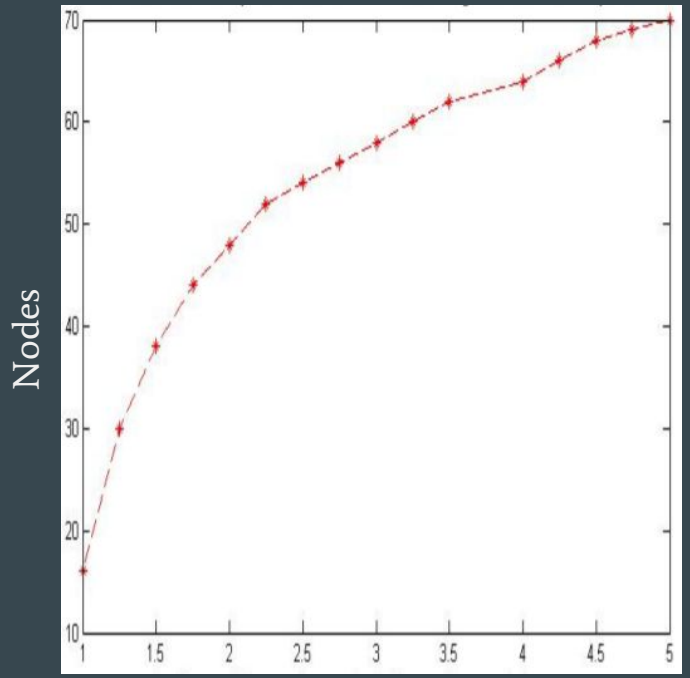
End Game

Information and search in computer chess

Alexandru Godescu, ETH Zurich 2011

- Humans look at certain pieces first to find the best moves
- Using information theory we can choose better nodes and depth to search
- These two things are proven to have very similar results

Decrease of importance of information moves vs plies



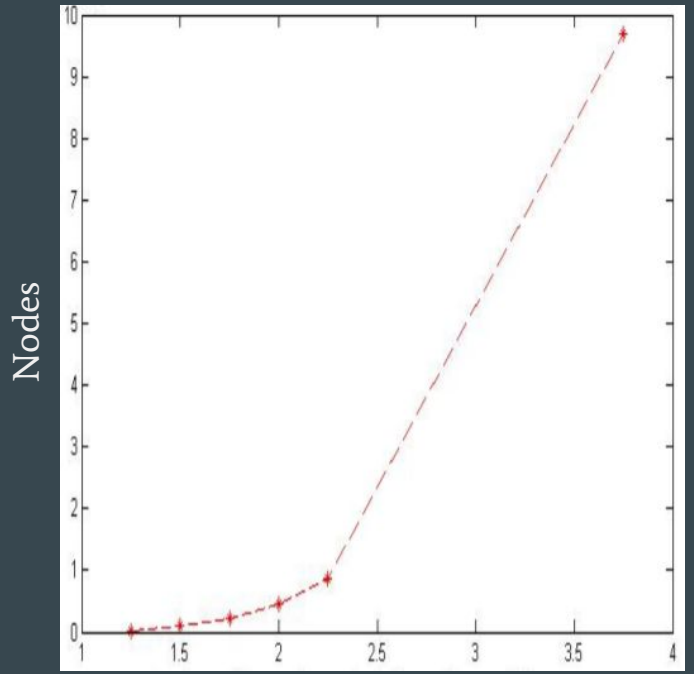
Reduction of importance to information gaining moves

Information and search in computer chess

Alexandru Godescu, ETH Zurich 2011

- Humans look at certain pieces first to find the best moves
- Using information theory we can choose better nodes and depth to search
- These two things are proven to have very similar results

Decrease of importance of information moves vs nodes increase



Reduction of importance to information gaining moves

Conclusion

- Current chess engines are beyond human players
- Recent advancements in Engines relate largely to improving evaluation of the board
- Chess Engines use more than Alpha-Beta
- Alpha-Beta is the most effective method for chess and the one used in the engines

Questions?

- K Greer, Tree Pruning for new search techniques in computer games, *Advances in Artificial Intelligence,* 2013
- Alexandru Godescu, Information and search in computer chess, *ETH Zurich,* 2011
- Diogo R. Ferreira, The impact of search depth on chess playing strength, *ICGA,* 2013
- GMC Haworth, Á Rusz, Position Criticality in Chess Endgames, *Advances in Computer Games,* 2011