Agents that Search Together: Adversarial Search

Russell and Norvig: Chapter 5

CSE 240: Winter 2023

Lecture 6

Announcements

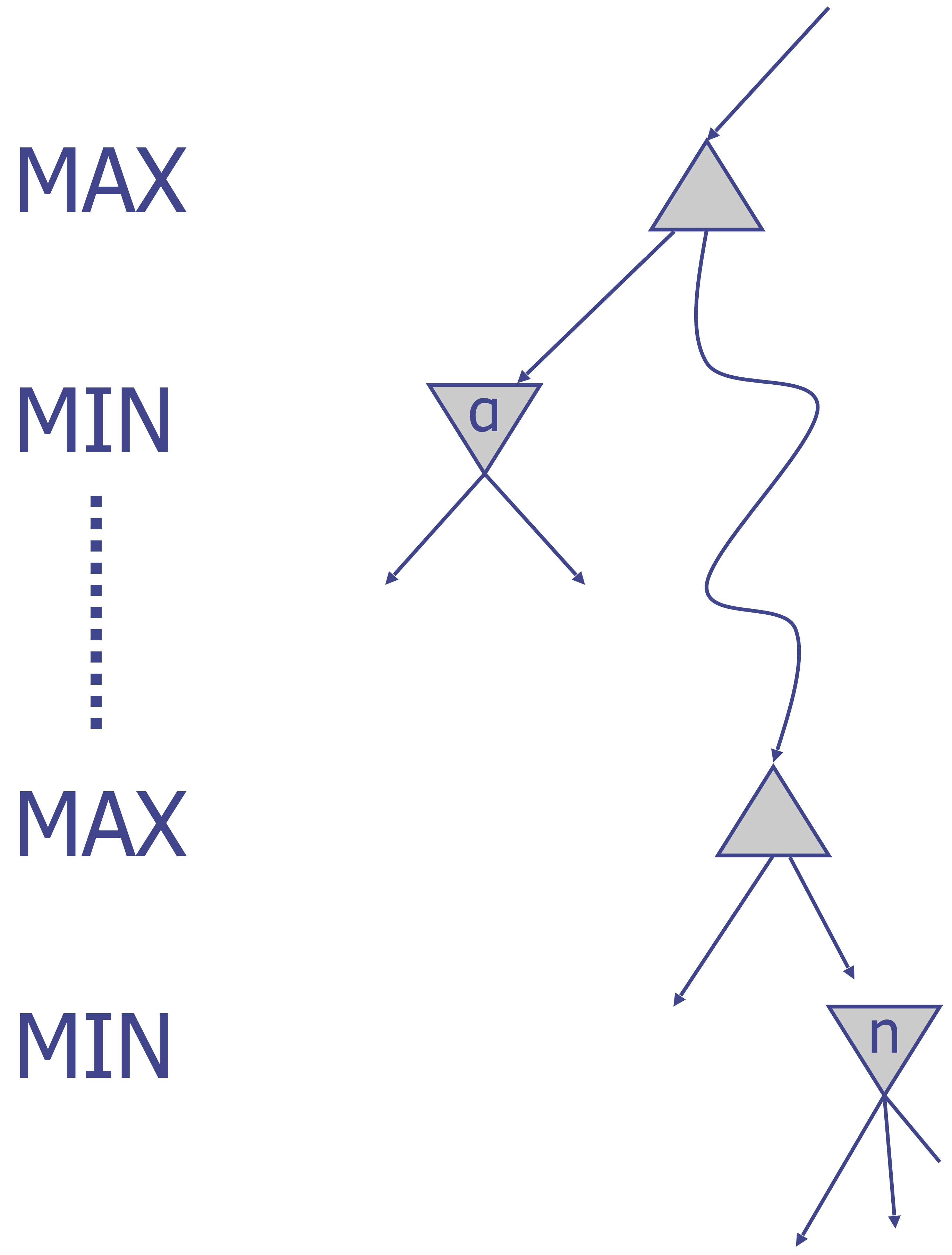
- Quiz 1 opens today after 11:25am
  - Due Friday at 5pm.
  - Open book, open note
  - 30 minutes
  - Time added for DRC.
- If you are using your late day, the assignment is due today at 5pm.
- Assignment 2 is posted.

Agenda

- Solving alpha-beta pruning example
- Handling resource limits
  - Alpha-beta pruning (last lecture)
  - Heuristic minimax algorithm
- Game agents in stochastic environments
- Break
- Quiz Review and Q/A

Alpha-Beta Pruning

- General configuration
  - We're computing the MIN-VALUE at n
  - We're looping over n's children
  - n's value estimate is dropping
  - α is the best value that MAX can get at any choice point along the current path
  - If n becomes worse than α, MAX will avoid it, so can stop considering n's other children
  - Define β similarly for MIN



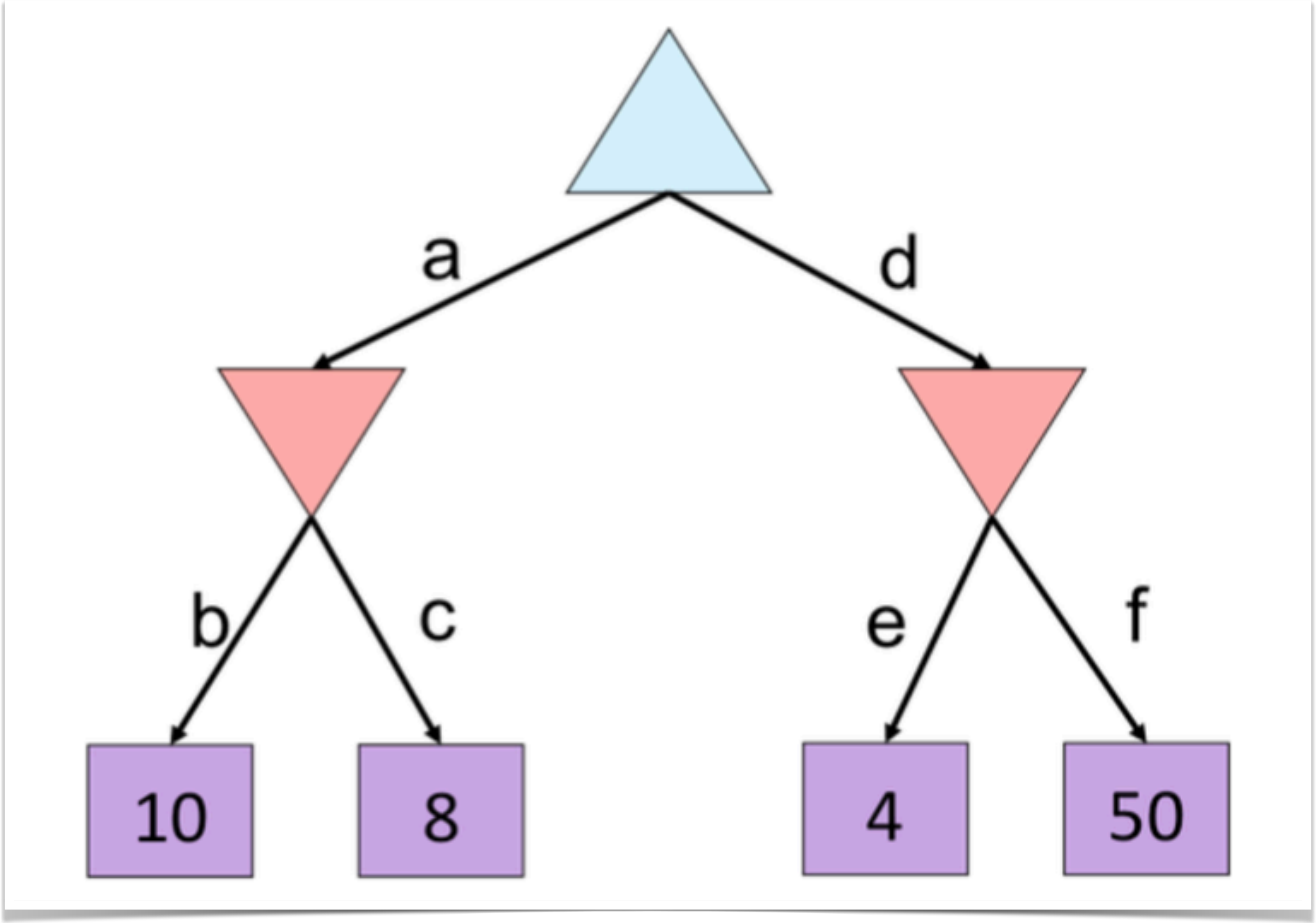
Alpha Beta Implementation

α: MAX's best option on path to root
β: MIN's best option on path to root

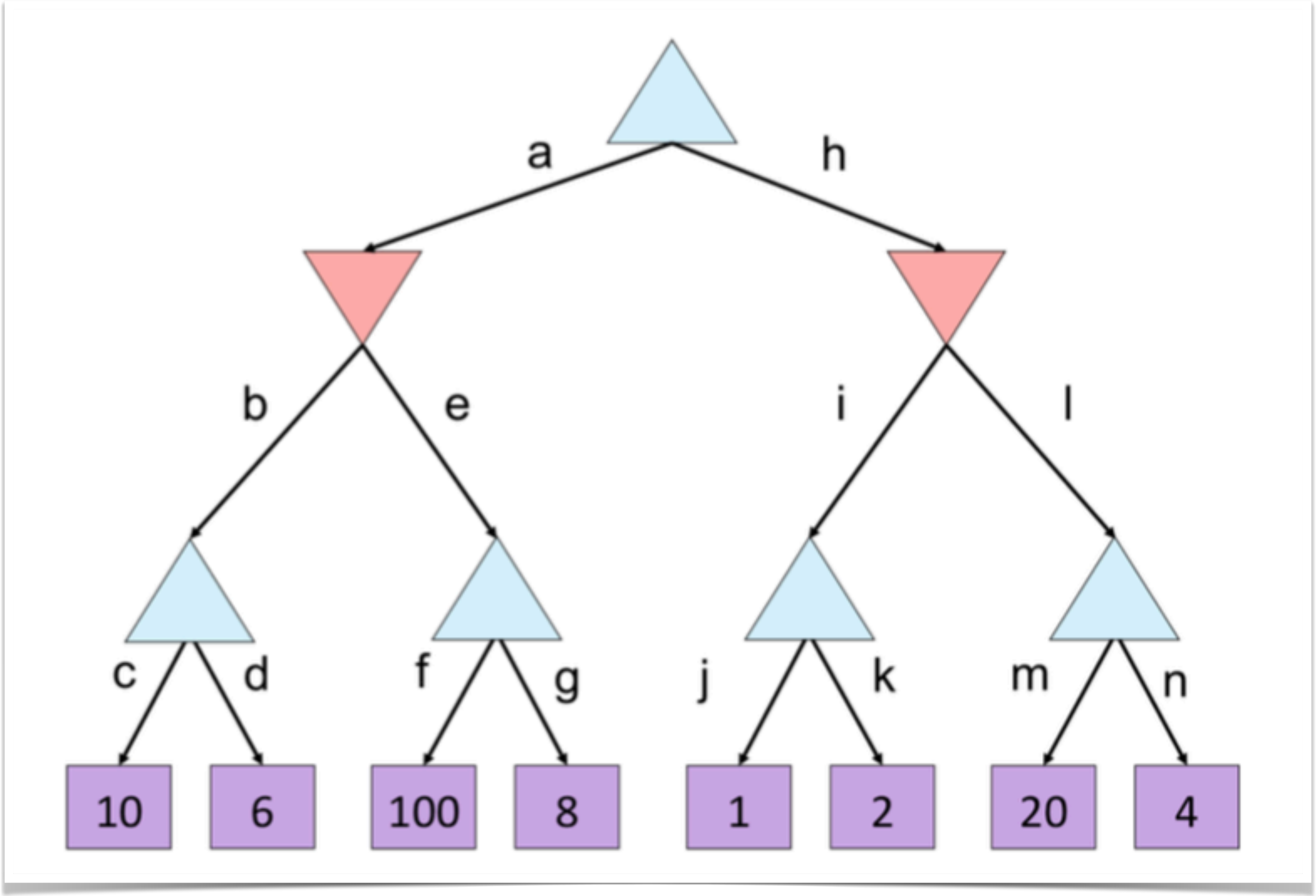
```
def max-value(state, α, β):
    initialize v = -∞
    for each successor of state:
        v = max(v, value(successor, α, β))
        if v ≥ β return v
        α = max(α, v)
    return v
```

```
def min-value(state , α, β):
    initialize v = +∞
    for each successor of state:
        v = min(v, value(successor, α, β))
        if v ≤ α return v
        β = min(β, v)
    return v
```

Alpha-Beta Practice 1



Alpha-Beta Practice 2

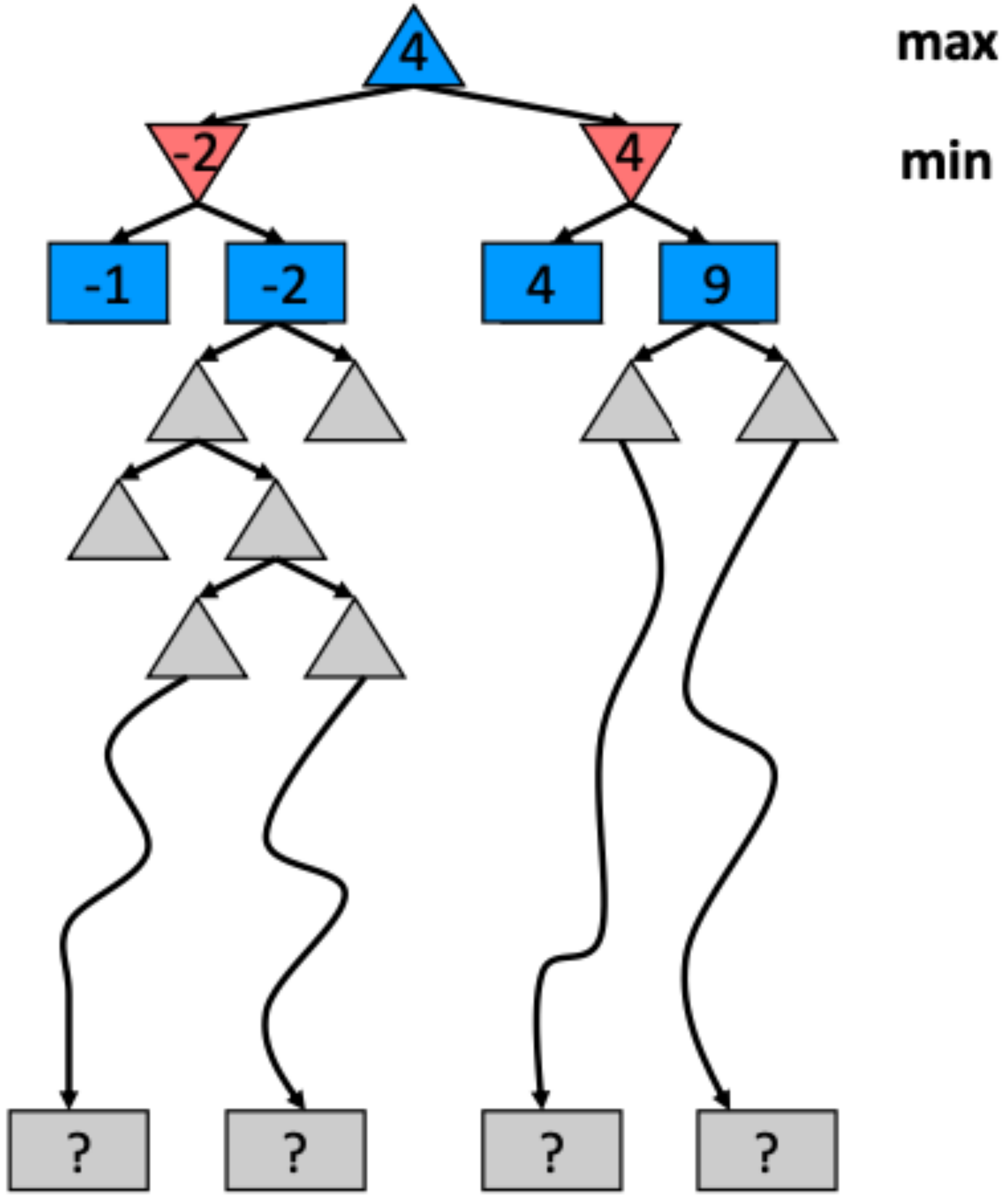


Resource Limits

Resource Limits

Problem: In realistic games, cannot search to leaves!

- Solution: Depth-limited search
    - Instead, search only to a limited depth in the tree
    - Replace terminal utilities with an evaluation function for non-terminal positions
- Example
    - Suppose we have 100 seconds, can explore 10K nodes / sec
    - So can check 1M nodes per move
    - $\alpha - \beta$ reaches about depth 8- decent chess program
- Guarantee of optimal play is gone.
- More plies make a BIG difference
- Use iterative deepening for the algorithm.



Depth Matters

- Evaluation functions are always imperfect.
- The deeper in the tree the evaluation is buried, the less the quality of the evaluation function matters.
- An important example of the tradeoff between complexity of features and complexity of computation

Heuristic Minimax

$$\boldsymbol{H_{MINIMAX}(s,d)}$$

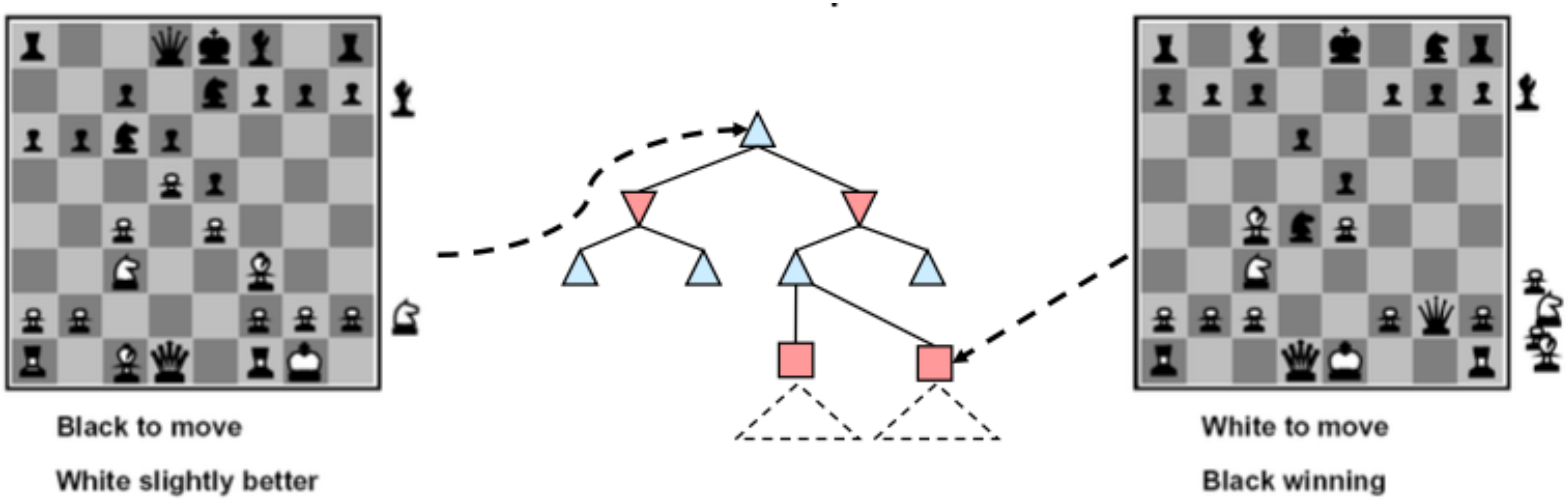
$$= \begin{cases} EVAL(s, MAX) & if\ CUTOFF\_TEST(s,d) \\ \max_{a \in ACTIONS(s)} H_{MINIMAX}(RESULT(s,a), d+1) & PLAYER(s) = MAX \\ \min_{a \in ACTIONS(s)} H_{MINIMAX}(RESULT(s,a), d+1) & PLAYER(s) = MIN \end{cases}$$

Evaluation Functions

Evaluation Functions

- For terminal states it should order them in the same way as the true utility function
- For non-terminal states, it should be strongly correlated with the actual chance of winning.
- It must not need high computational cost.
- Use features for calculating the evaluation function, such as king safety, good pawn structure, etc.

Evaluation Functions



- Evaluation functions score non-terminals in depth-limited search
- Ideal function: returns the actual minimax value of the position
- In practice: typically weighted linear sum of features:
  $Eval(s) = w_1 f_1(s) + w_2 f_2(s) + \ldots + w_n f_n(s)$
- e.g. $f_1(s)$ = (num white queens - num black queens, etc.

5 minute break

Quiz Review

Quiz Outline

- 7 questions, 30 minutes (+ DRC)
- Questions
  - Definition of rationality
  - Time and space complexity
  - Alpha beta definitions
  - Search trees
  - Heuristics
  - Minimax
  - Alpha beta pruning

Rationality

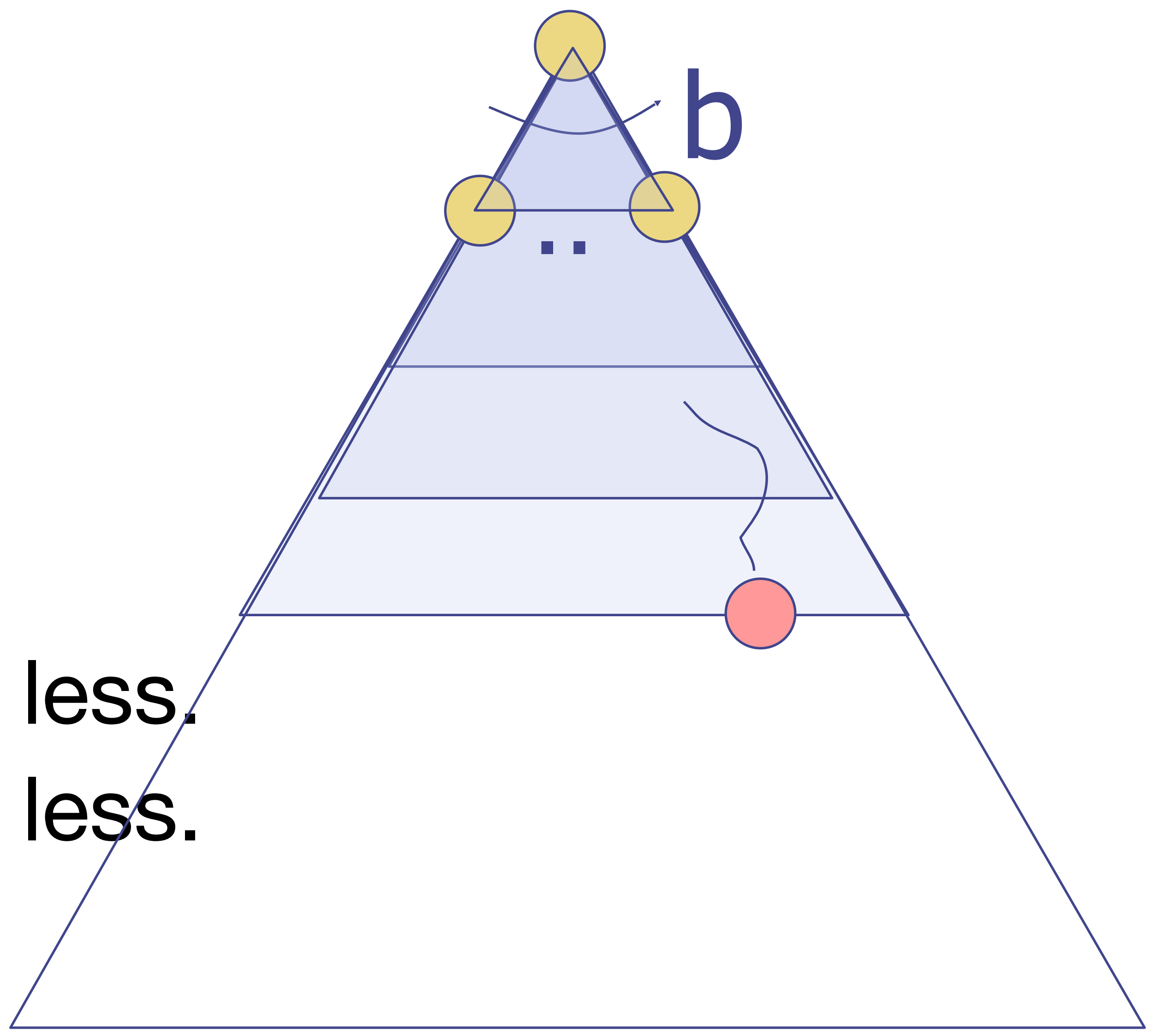
- Rationality means doing the right thing.
- Philosophers -> mind is in some ways like a machine and it operates based on the encoded knowledge.
- Mathematicians -> provided tools to use logical statements for reasoning and decision making.
- Economics -> formalized the problem of decision making using maximization of expected outcome.

Rational Agent

What is rational at any given time depends on four things:	1. The performance measure (agent function) that defines the criterion of success 2. The agent’s prior knowledge of the environment 3. The actions that the agent can perform 4. The agent’s percept sequence to date.

Definition	For each possible percept sequence, a rational agent should select an action that is expected to maximize its performance measure given the evidence provided by the percept sequence and whatever built-in knowledge the agent has,

Iterative Deepening



Iterative deepening uses DFS as a subroutine:

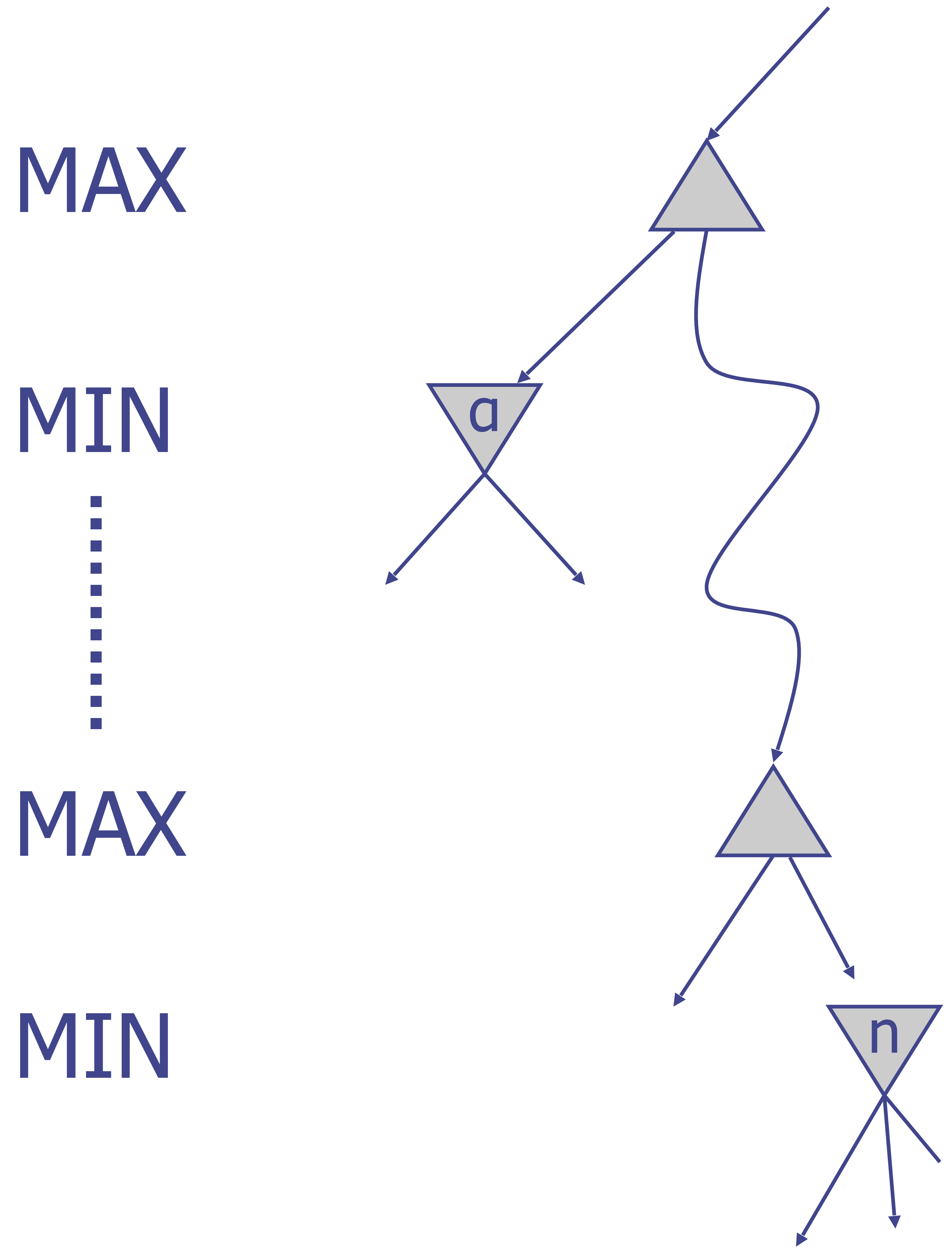
1. Do a DFS which only searches for paths of length 1 or less.
2. If “1” failed, do a DFS which only searches paths of length 2 or less.
3. If “2” failed, do a DFS which only searches paths of length 3 or less.

….and so on.

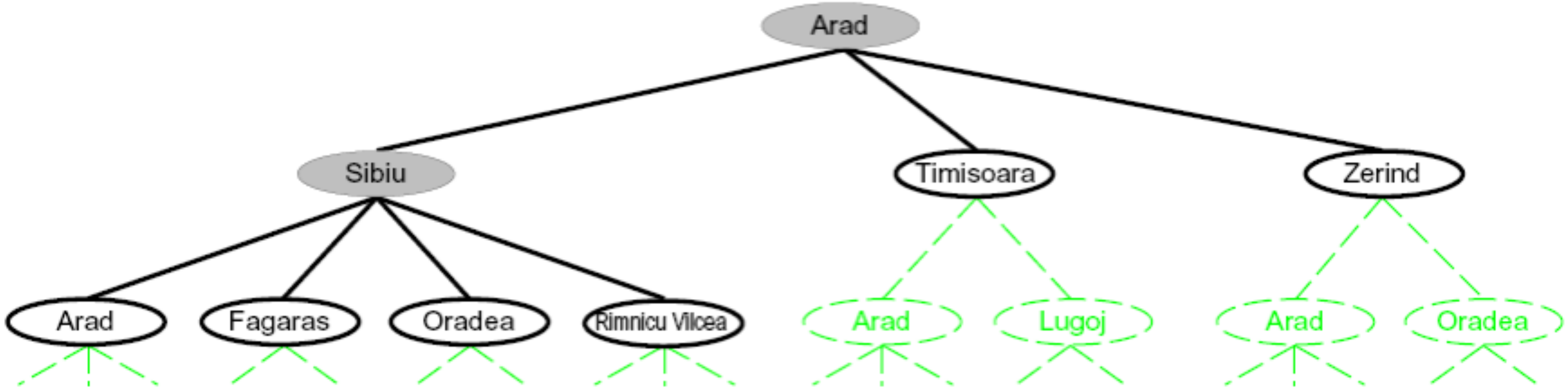
Algorithm		Complete	Optimal	Time	Space
DFS	w/ Path Checking	Y	N	$O(b^{m+1})$	$O(bm)$
BFS		Y	N*	$O(b^{s+1})$	$O(b^s)$
ID		Y	N*	$O(b^{s+1})$	$O(bs)$

Alpha-Beta Pruning

- General configuration
  - We're computing the MIN-VALUE at n
  - We're looping over n's children
  - n's value estimate is dropping
  - α is the best value that MAX can get at any choice point along the current path
  - If n becomes worse than α, MAX will avoid it, so can stop considering n's other children
  - Define β similarly for MIN



Another Search Tree



- Search:
  - Expand out possible plans
  - Maintain a fringe of unexpanded plans
  - Try to expand as few tree nodes as possible

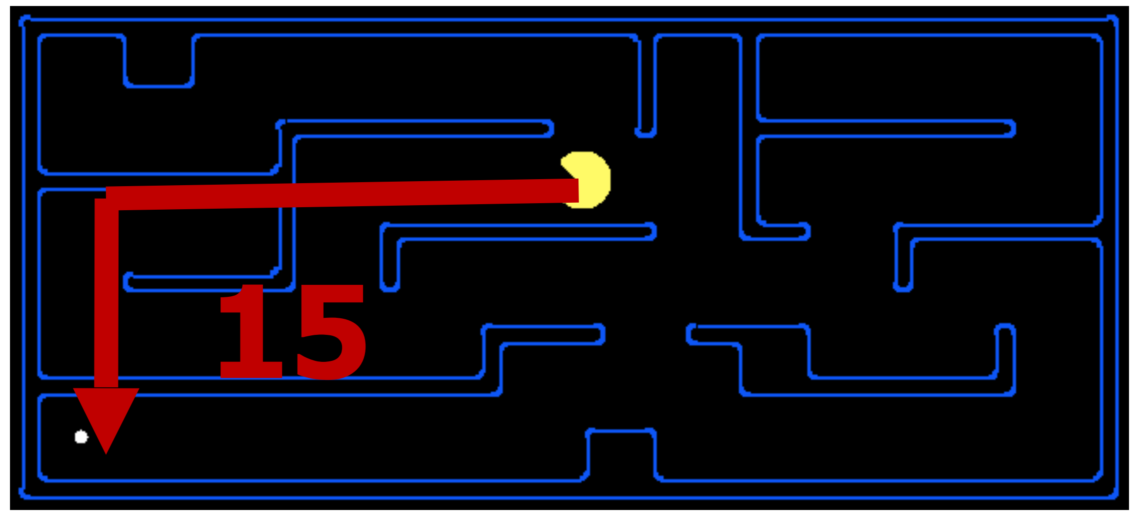
Admissible Heuristics

- A heuristic h is admissible (optimistic) if:

$$h(n) \leq h^*(n)$$

- where $h^*(n)$ is the true cost to a nearest goal

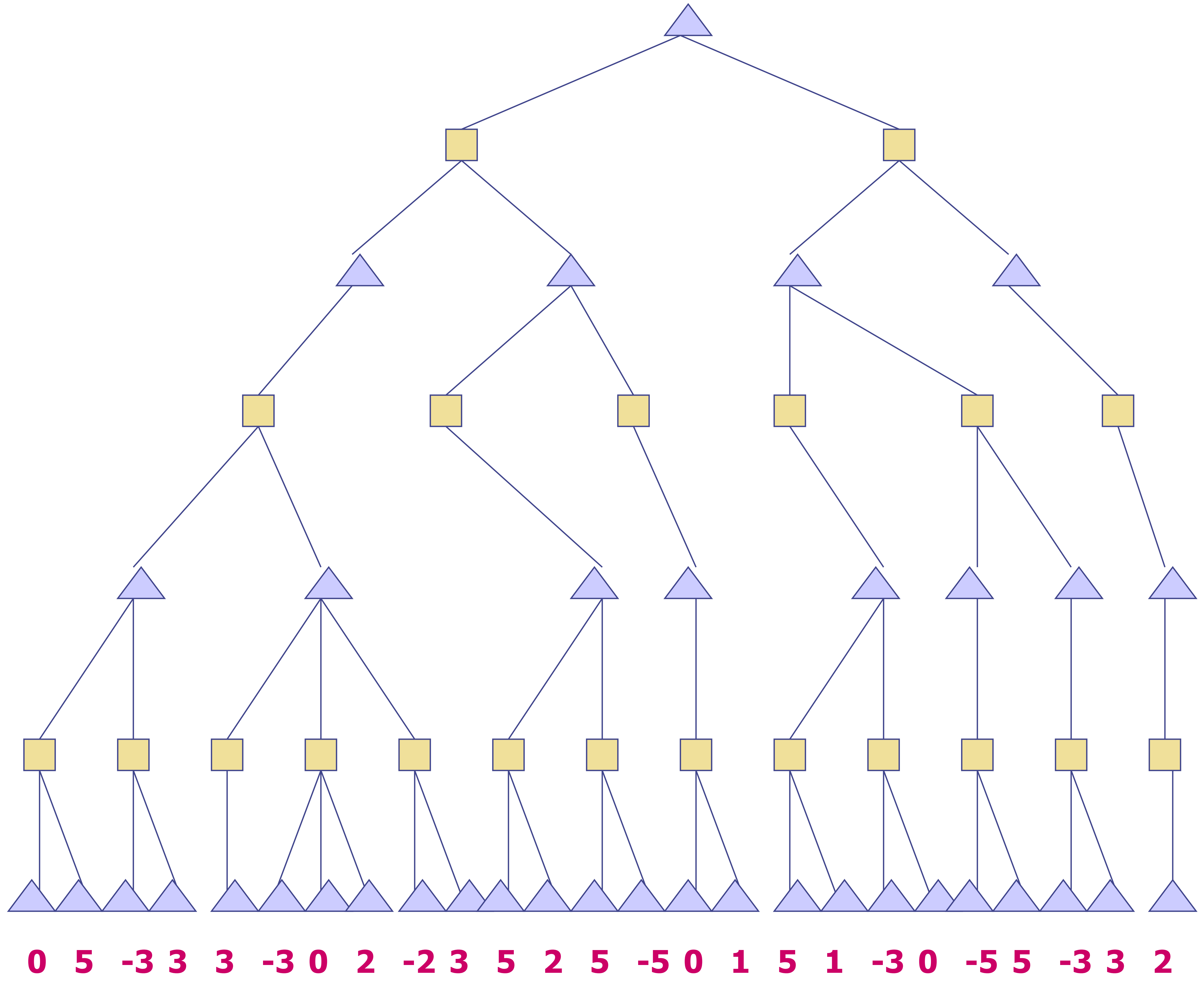
- Example:



- Coming up with admissible heuristics is most of what's involved in using A* in practice.

Nice Video: https://www.youtube.com/watch?v=xBXHtz4Gbdo

Alpha-Beta Example



Recap

- Game theory
  - Adversarial games
    - Minimax algorithm and alpha-beta pruning
- Next week
  - Stochastic games
    - Expectimax search algorithm