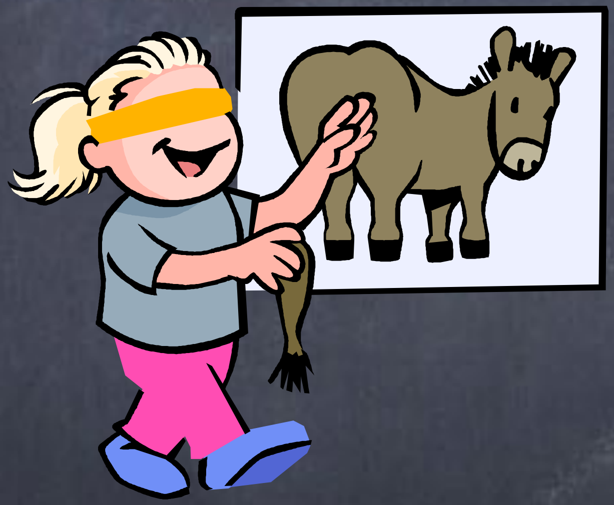
CSC242:
Introduction to Artificial Intelligence

Lecture 1.3

Please put away all electronic devices

Search Strategies

Uninformed



No additional information about states

Informed (Heuristic)

Can identify "promising" states

Search Strategies

Uninformed

	BFS	DFS (tree)	IDS	Greedy	A*	IDA*
Complete ?	✓	✗	✓	✗	✓†	✓†
Optimal ?	✓*	✗	✓*	✗	✓†	✓*†
Time	$O(b^d)$	$O(b^m)$	$O(b^d)$	$O(b^m)$	$O(b^{\epsilon d})$	$O(b^{\epsilon d})$
Space	$O(b^d)$	$O(bm)$	$O(bd)$	$O(b^m)$	$O(b^d)$	$O(bd)$

* If step costs are identical

† With an admissible heuristic

Search Strategies

Uninformed Informed

	BFS	DFS (tree)	IDS	Greedy	A*	IDA*
Complete?	✓	✗	✓	✗	✓†	✓†
Optimal?	✓*	✗	✓*	✗	✓†	✓*†
Time	$O(b^d)$	$O(b^m)$	$O(b^d)$	$O(b^m)$	$O(b^{\epsilon d})$	$O(b^{\epsilon d})$
Space	$O(b^d)$	$O(bm)$	$O(bd)$	$O(b^m)$	$O(b^d)$	$O(bd)$

* If step costs are identical

† With an admissible heuristic

Evaluation function

$$f(n) = g(n) + h(n)$$

True cost of path from start node to node n

Estimated cost of cheapest path from n to a goal node

Admissible Heuristic

Never overestimates the true cost of a solution

$$f(n) = g(n) + h_{SLD}(n)$$

Search Strategies

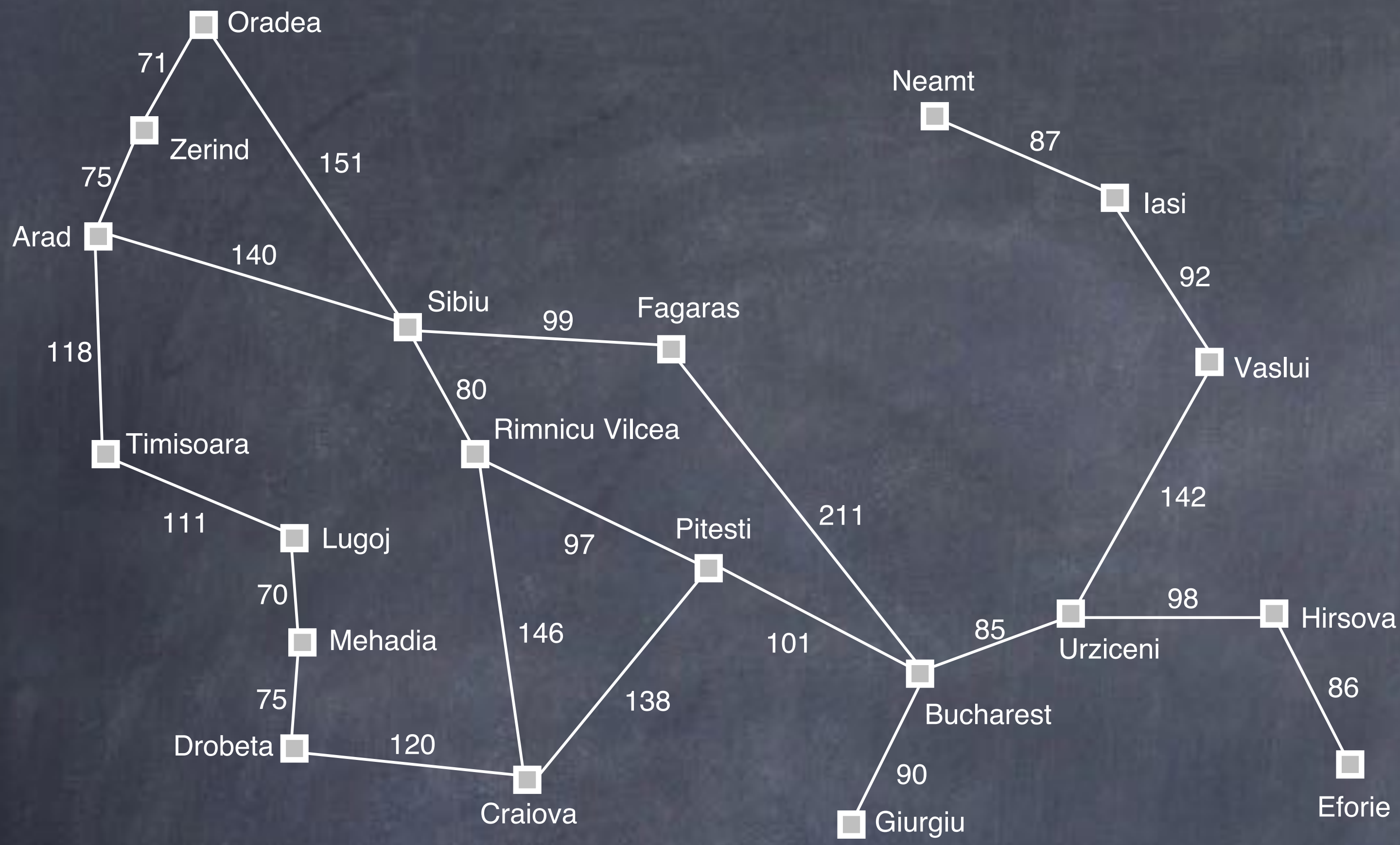
Uninformed | Informed

	BFS	DFS (tree)	IDS	Greedy	A*	IDA*
Complete ?	✓	✗	✓	✗	✓†	✓†
Optimal ?	✓*	✗	✓*	✗	✓†	✓*†
Time	$O(b^d)$	$O(b^m)$	$O(b^d)$	$O(b^m)$	$O(b^{\epsilon d})$	$O(b^{\epsilon d})$
Space	$O(b^d)$	$O(bm)$	$O(bd)$	$O(b^m)$	$O(b^d)$	$O(bd)$

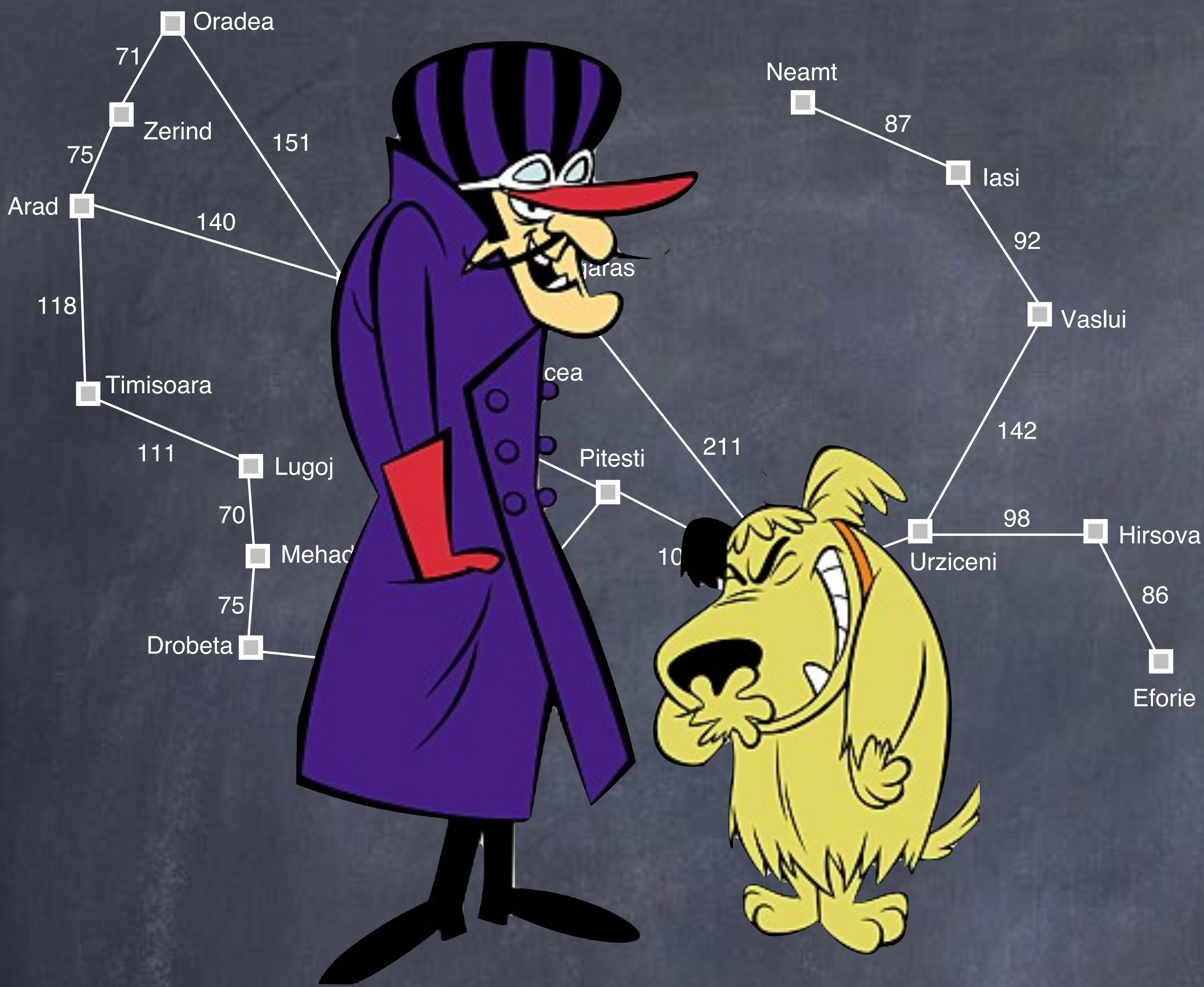
* If step costs are identical

† With an admissible heuristic

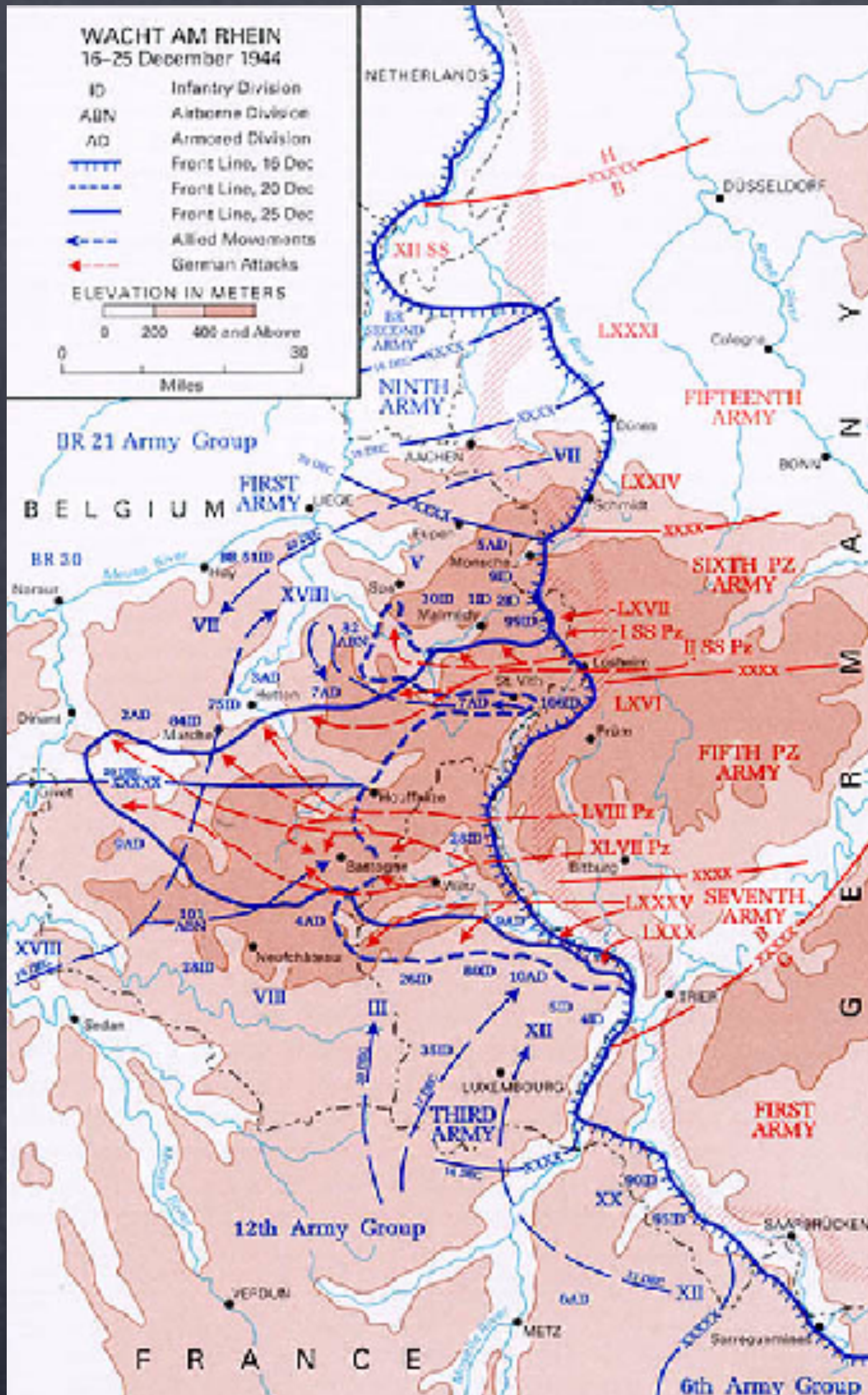
Adversarial Search


Oradea
71
Zerind
151
75
Arad
140
Sibiu
99
Fagaras
118
80
Rimnicu Vilcea
Timisoara
111
Lugoj
97
Pitesti
211
70
Mehadia
146
101
75
138
Drobeta
120
Craiova
Neamt
87
Iasi
92
Vaslui
142
98
Hirsova
85
Urziceni
86
Bucharest
90
Giurgiu
Eforie


ThinkFun
Fifteen Puzzle
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15


Oradea
71
Zerind
151
75
Arad
140
118
Timisoara
111
Lugoj
70
75
Drobeta
Pitesti
211
Neamt
87
Iasi
92
Vaslui
142
98
Urziceni
Hirsova
86
Eforie


ThinkFun
Fifteen Puzzle



Battle of The Bulge

16-25 Dec 1944

Competitive Landscape

User Needs

Design

Product

Marketing

Support

Multi-Agent Environments

- Unpredictability of other agents => contingencies (strategies)
- Agents goals in conflict => competition

Games

Games

Games

- "Games require the ability to make some decision even when calculating the optimal decision is infeasible"
- "[Games] penalize inefficiency severely"

Games $\neq$ Toy Problems

Games ≠ Toy Problems

8-puzzle: 181,440

15-puzzle: ~1.3×10^{12}

24-puzzle: ~10^{25}

Games ≠ Toy Problems

8-puzzle: 181,440

15-puzzle: ~1.3×10^{12}

24-puzzle: ~10^{25}

$35^{100} = 10^{154}$

(only 10^{40} distinct)

Games $\neq$ Toy Problems

8-puzzle: 181,440

15-puzzle: $\sim 1.3 \times 10^{12}$

24-puzzle: $\sim 10^{25}$

$35^{100} = 10^{154}$

(only 10^{40} distinct)

2×10^{170}

Abstract Games

Abstract Games

- Deterministic (no chance)
- Nondeterministic (dice, cards, etc.)

Abstract Games

- Perfect information (fully observable)
- Imperfect information (partially observable)

Abstract Games

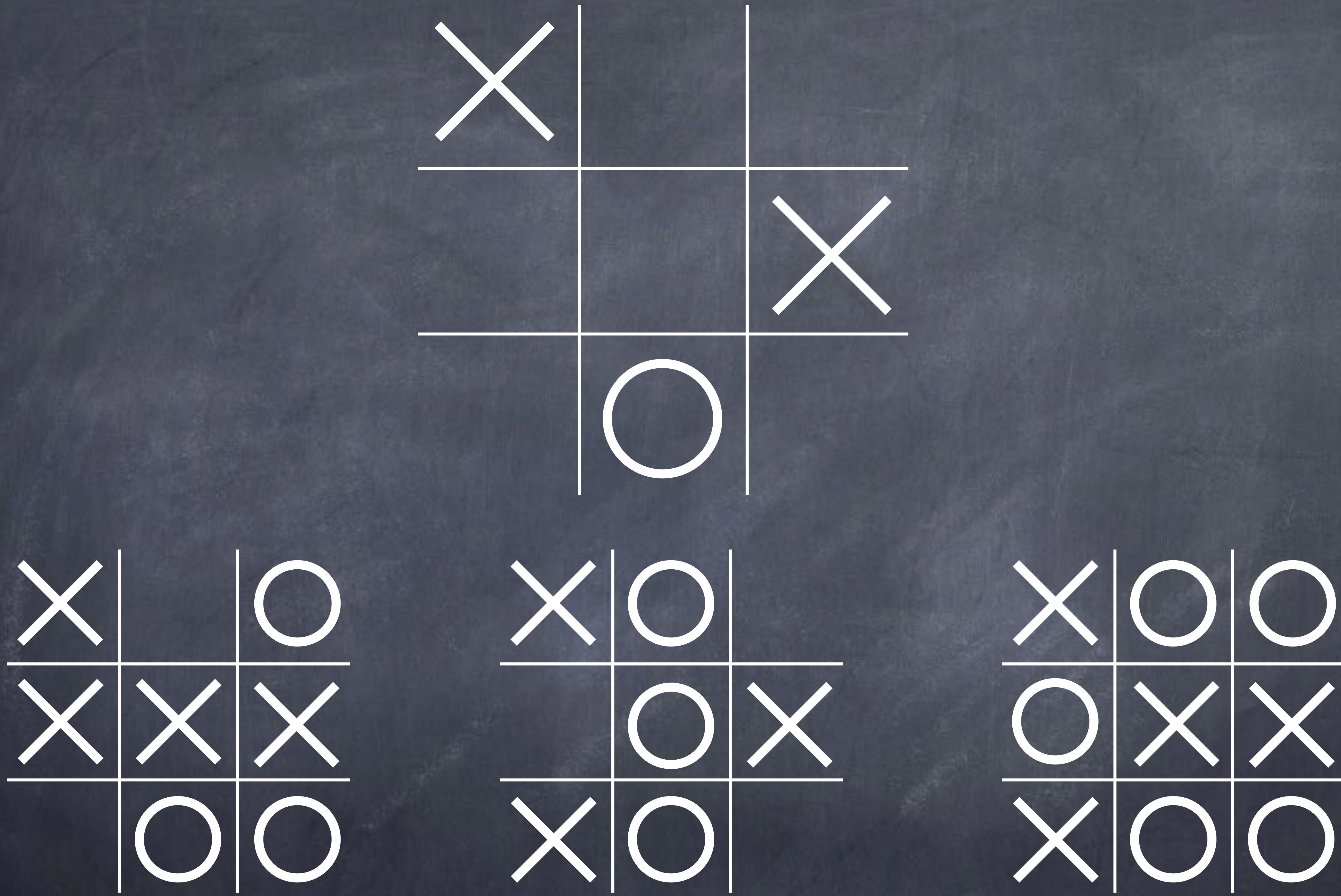
- Zero-sum (total payoff the same in any game)
  - What's good for me is bad for you and vice-versa
- Arbitrary utility function

Types of Games

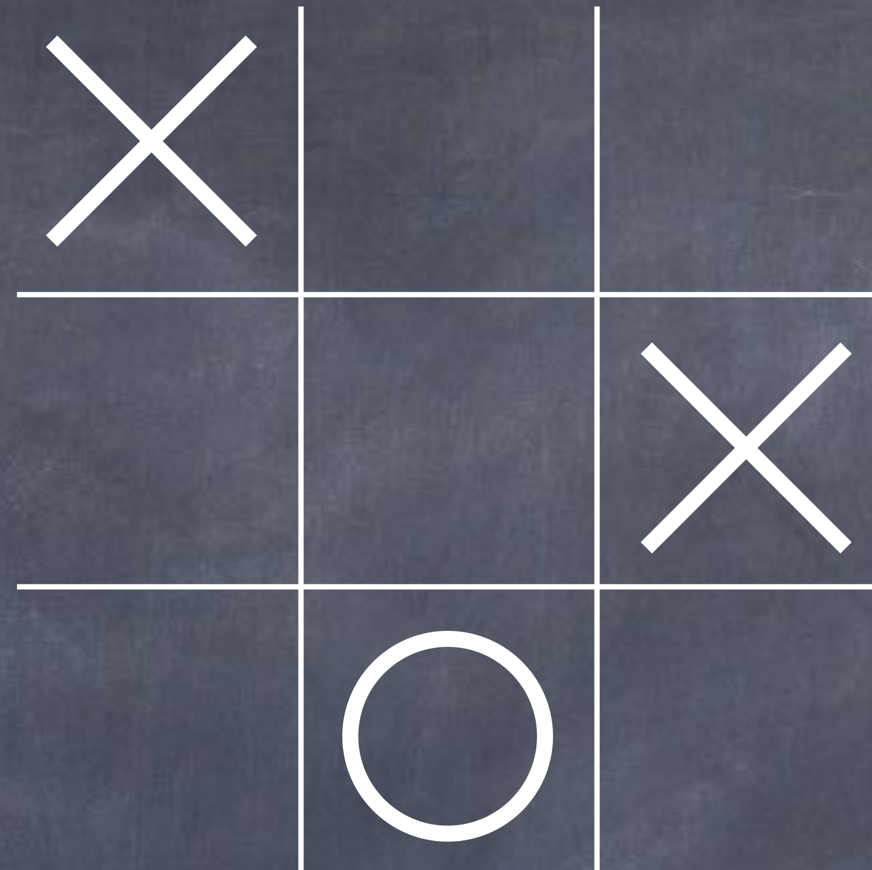
Deterministic (no chance)	Nondeterministic (dice, cards, etc.)
Perfect information (fully observable)	Imperfect information (partially observable)
Zero-sum (total payoff the same in any game)	Arbitrary utility functions

Outline

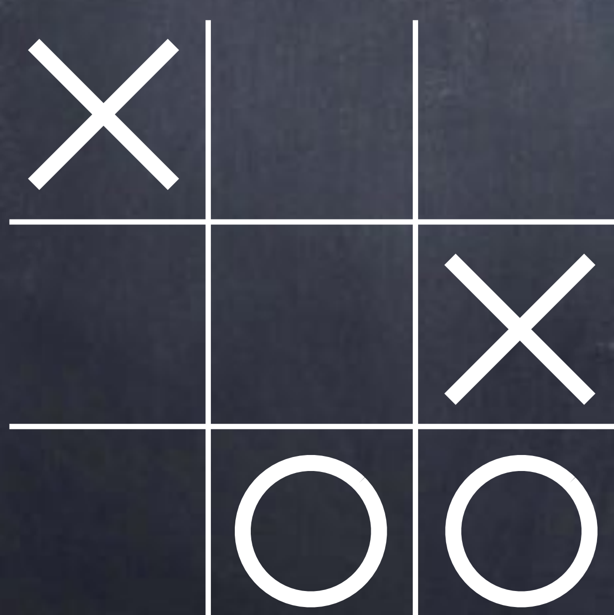
- In deterministic, perfect information, zero-sum games:
  - How to find optimal moves
  - How to find good moves when time is limited
- Next time: nondeterministic
- Later: imperfect information, utilities

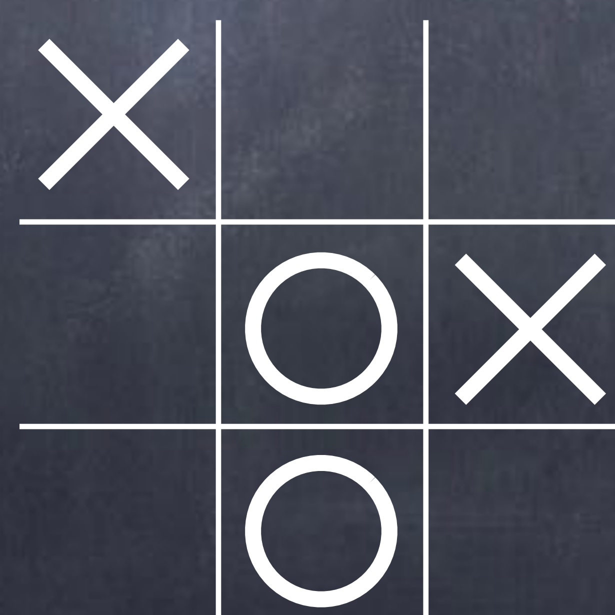


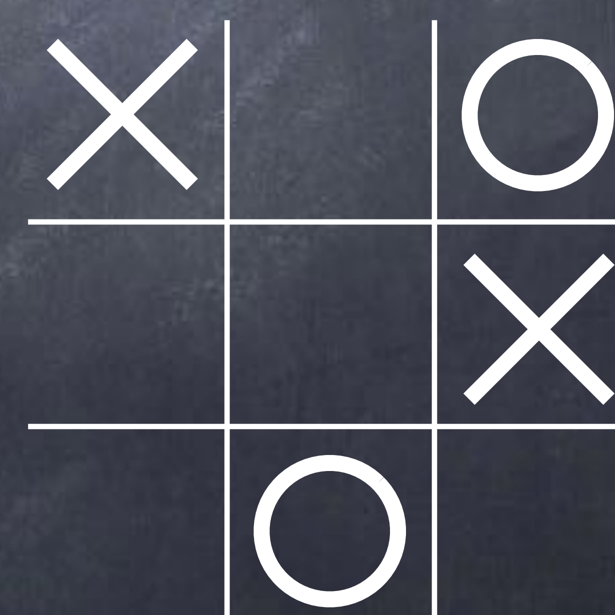
Terminal States

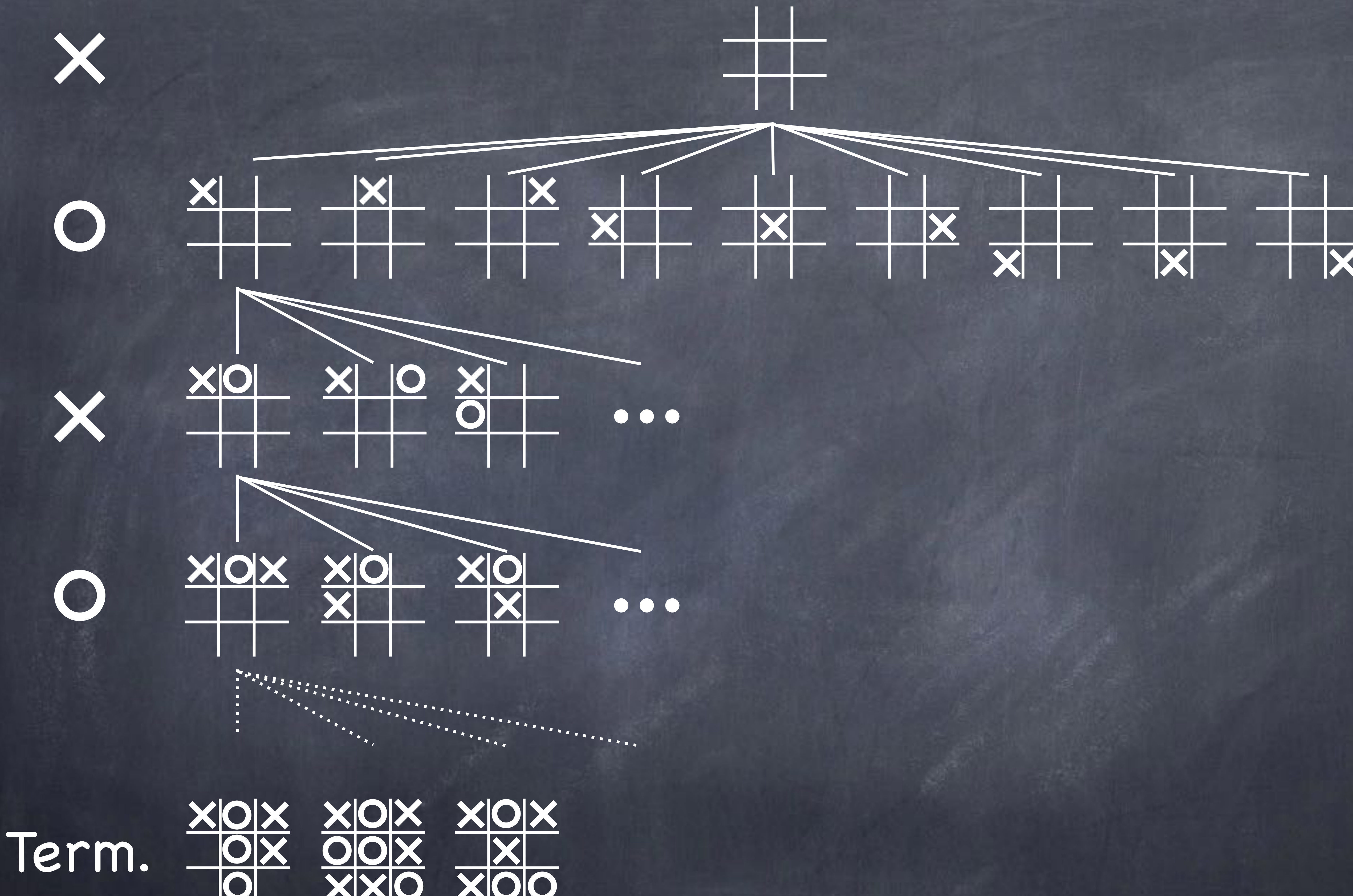


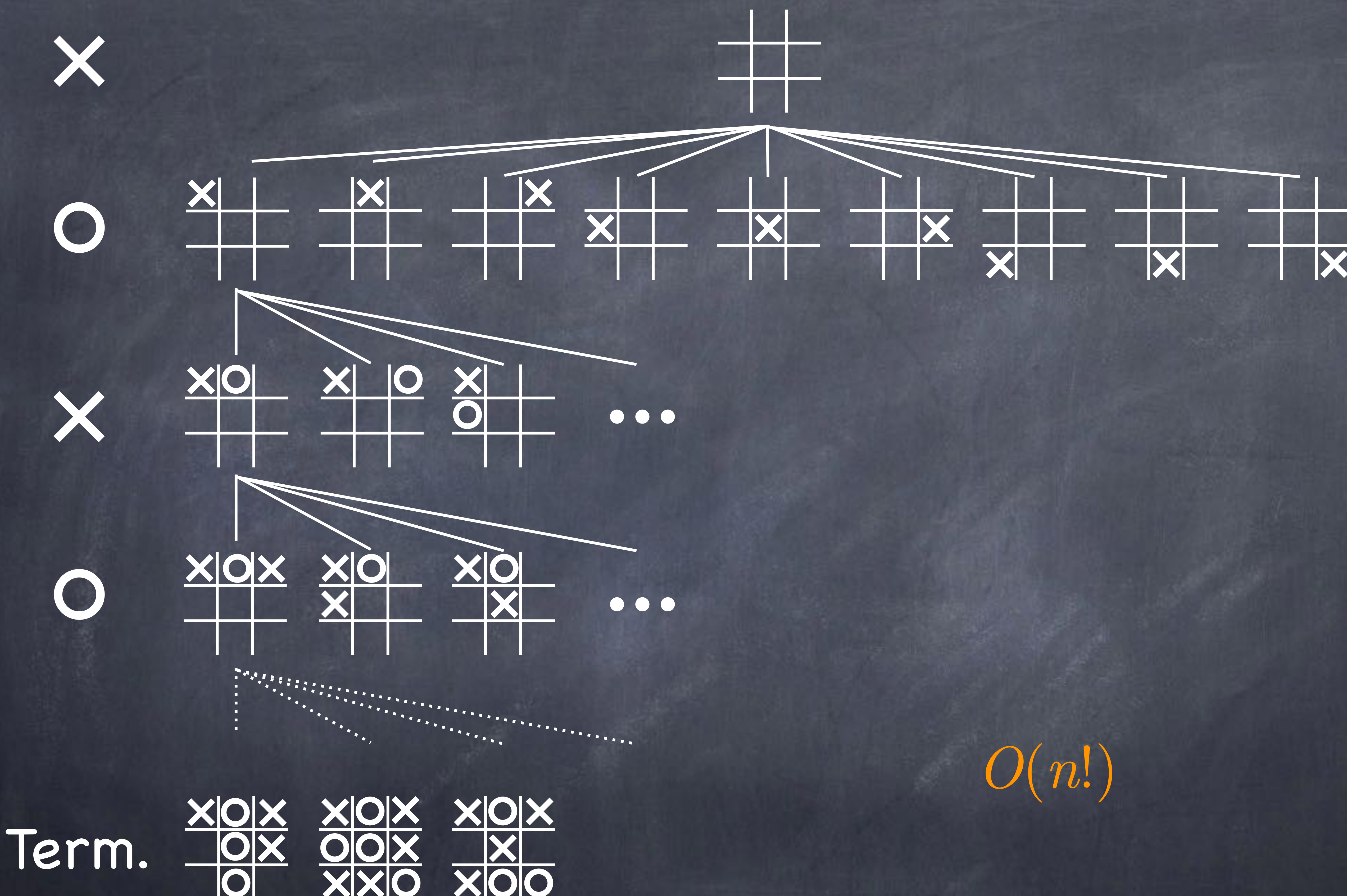
Next Player: O



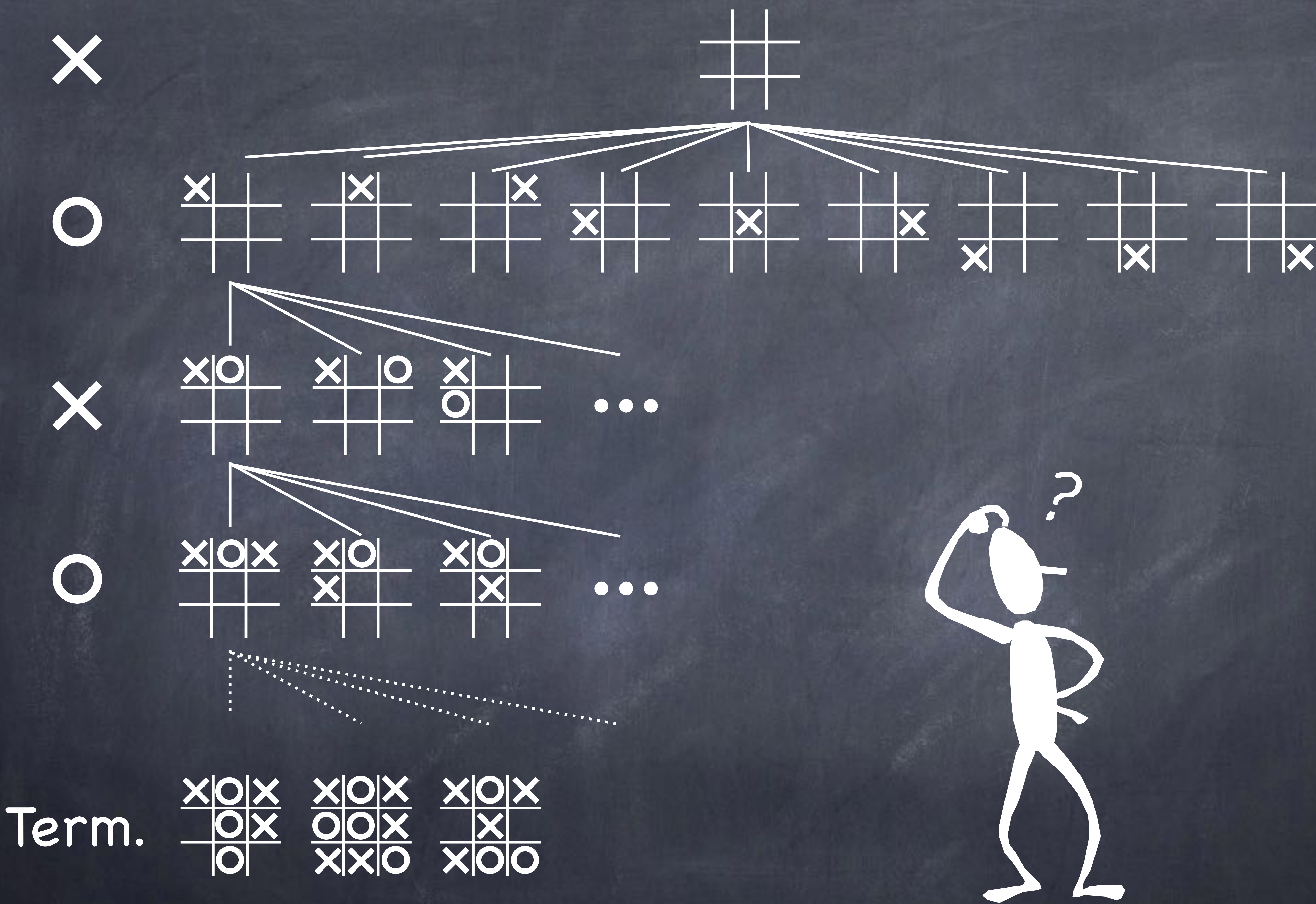





Term.

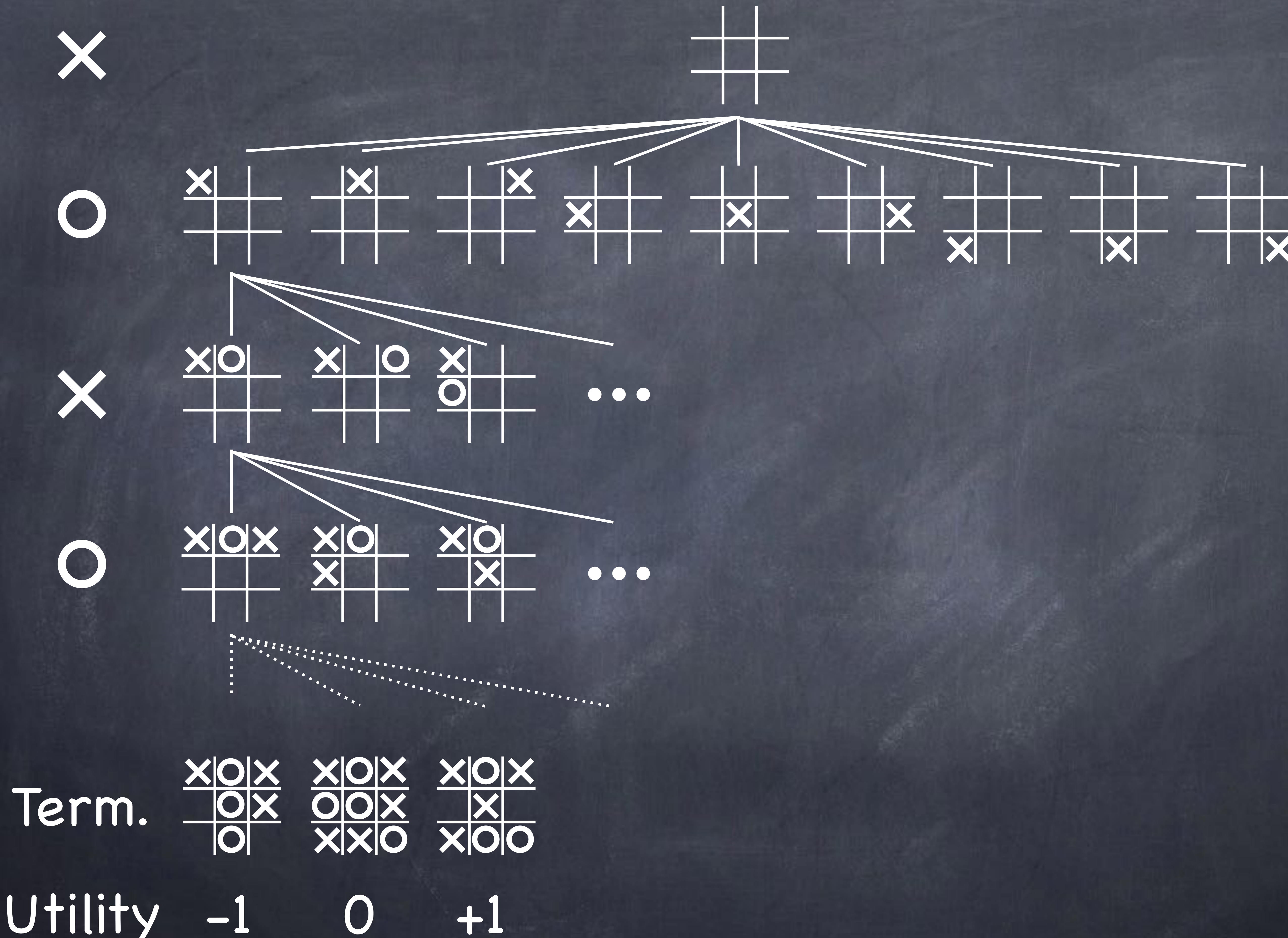


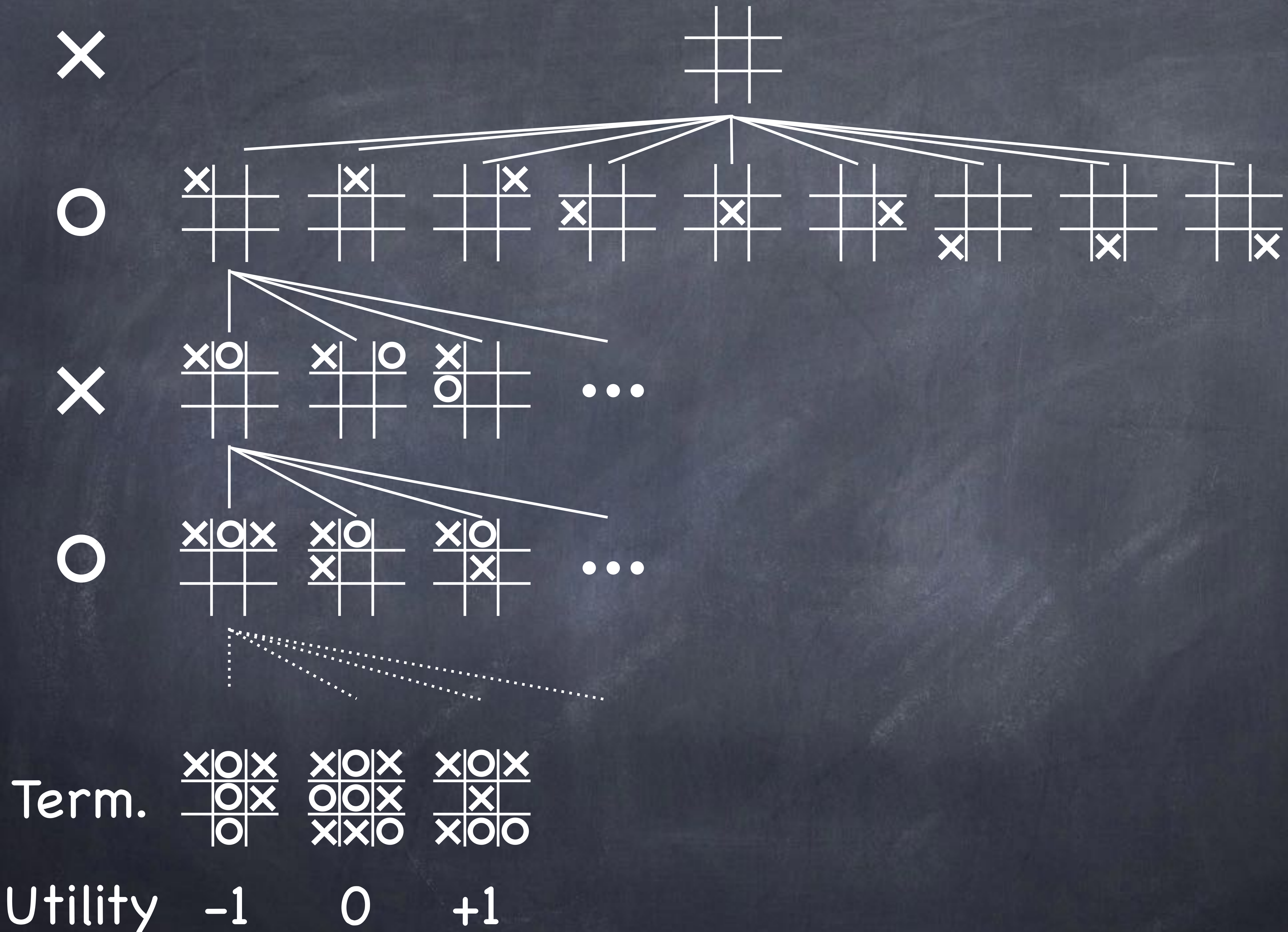
$O(n!)$

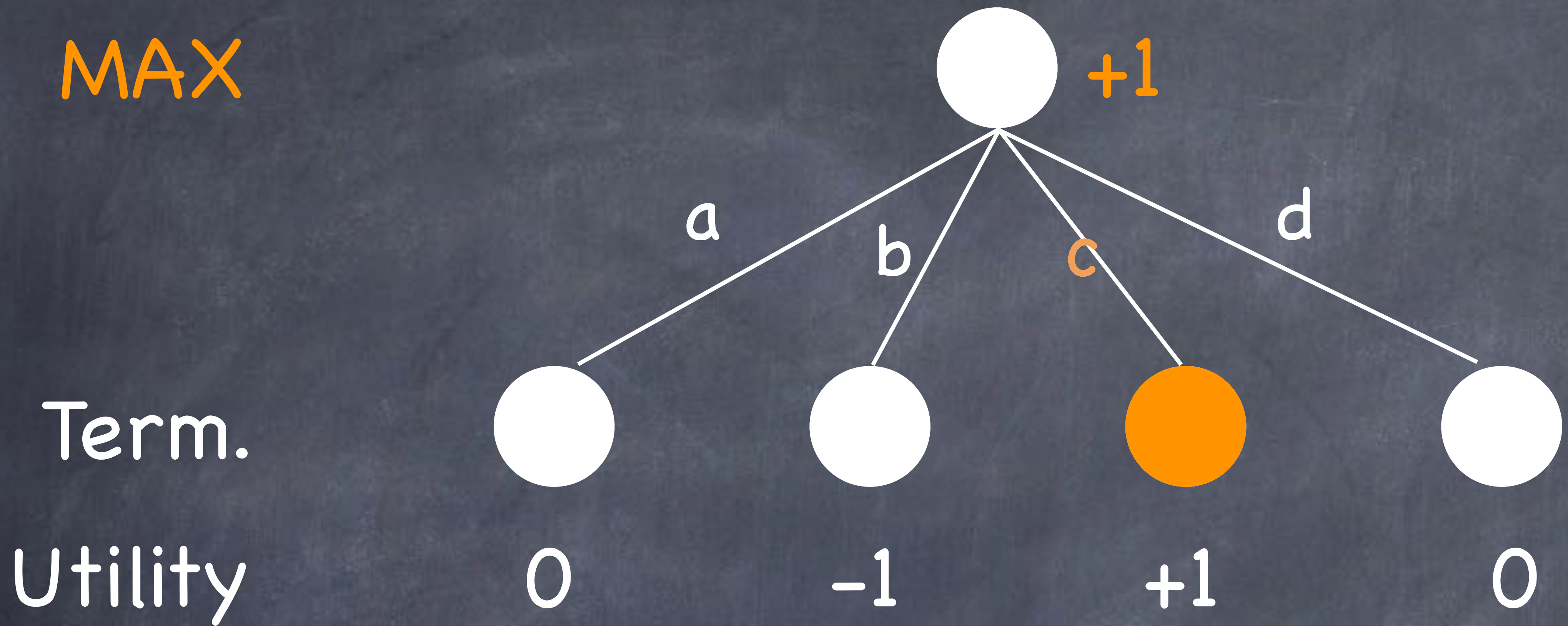

X
O
X
O
Term.

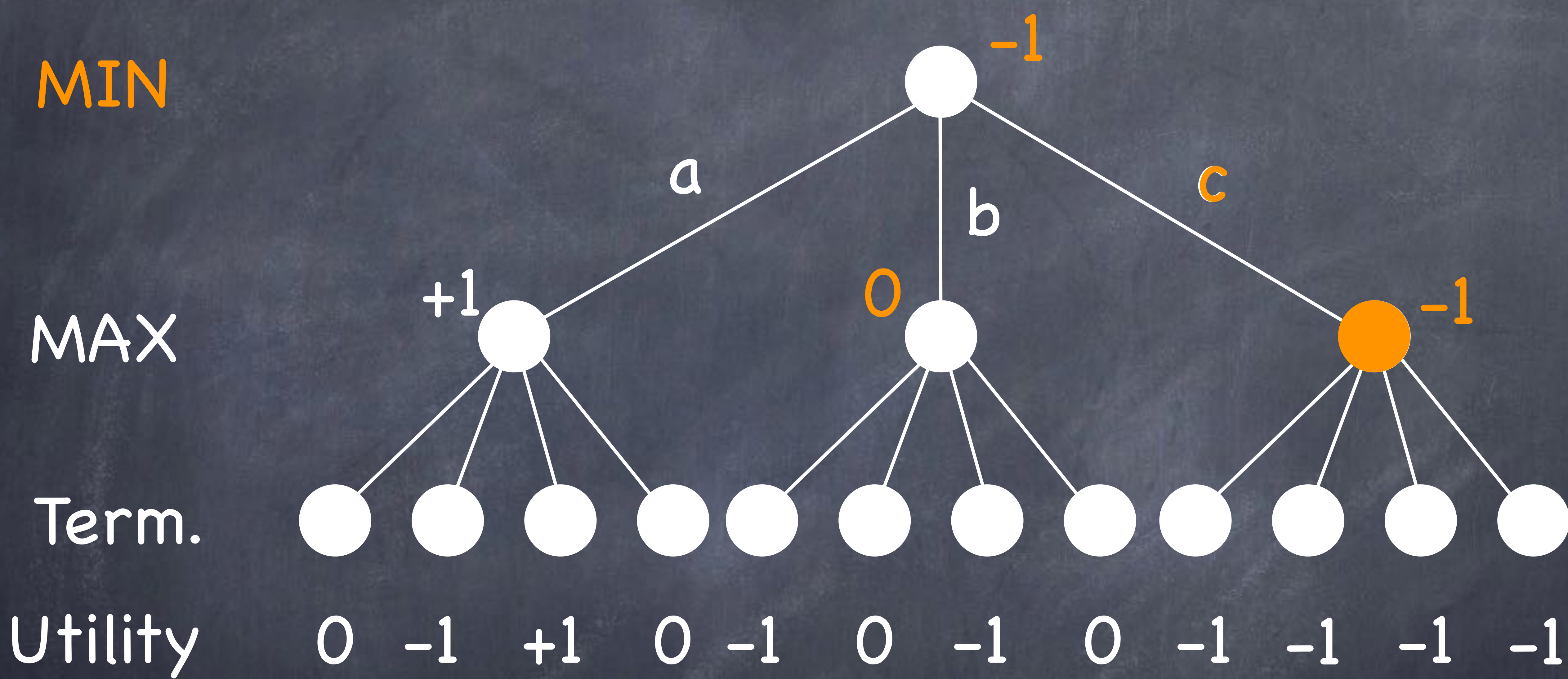
Deciding what to do in a game requires thinking about what the opponent will do, and having a strategy that takes that into account

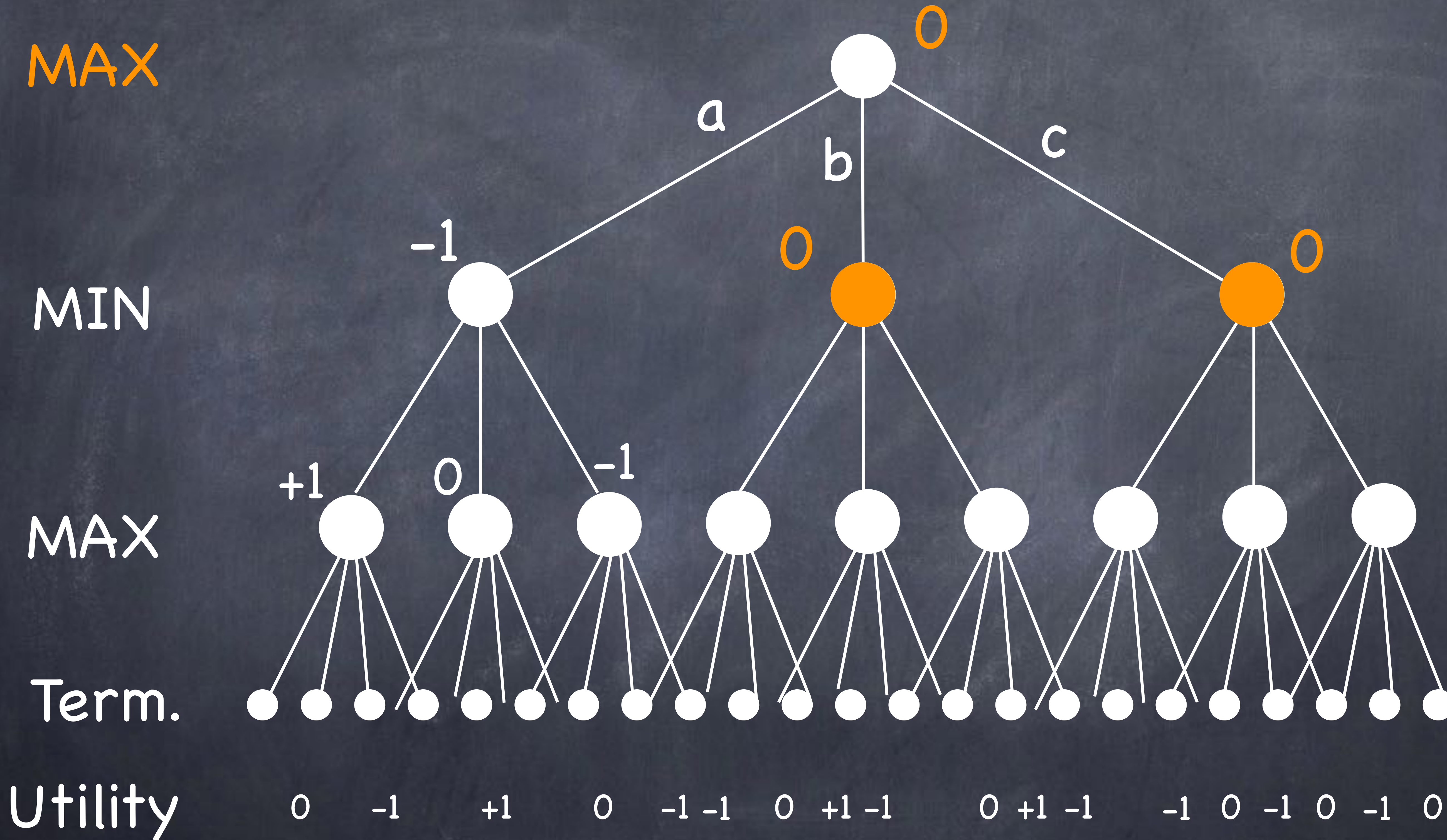
Not simply a sequence of actions, but a contingency plan that specifies what to do depending on what state one finds oneself in (AIMA 4.3)


X
O
X
O
Term.
Utility
–1
0
+1


X
O
X
O
Term.
Utility
−1
0
+1


MAX
+1
a
b
c
d
Term.
Utility
0
-1
+1
0

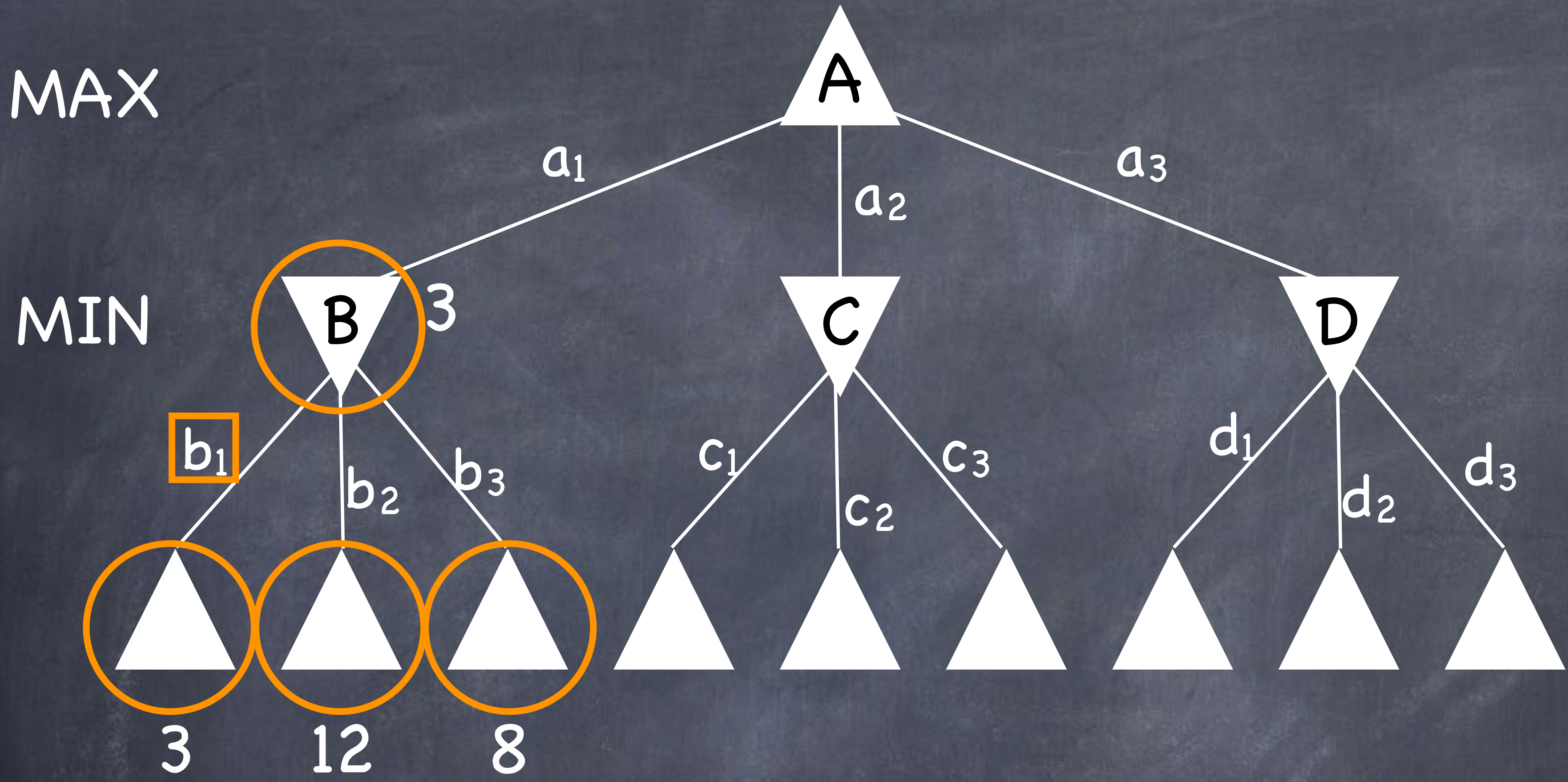

MIN
MAX
Term.
Utility
-1
a
b
c
+1
0
-1
0 -1 +1 0 -1 0 -1 0 -1 -1 -1 -1

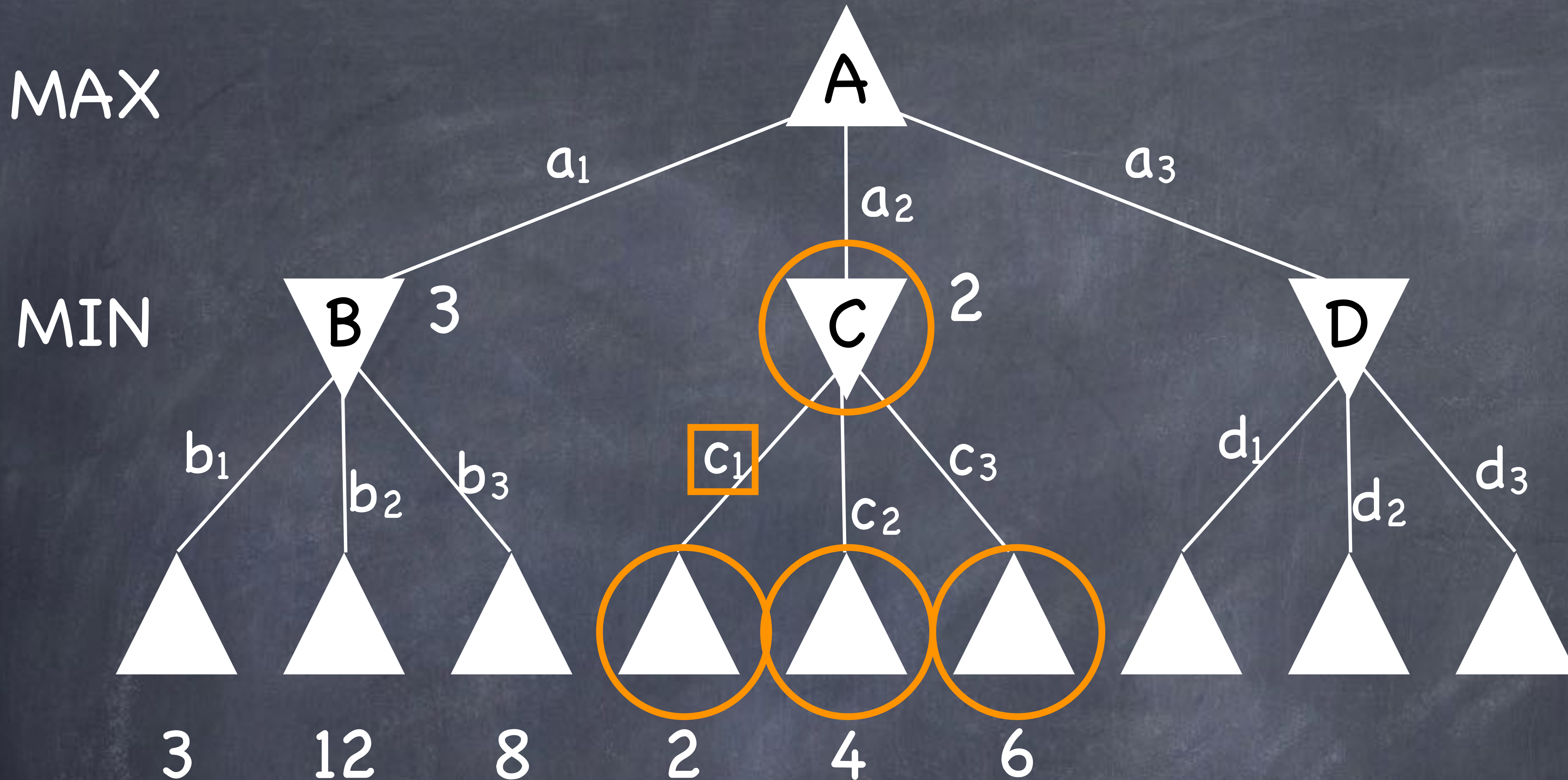


MAX
MIN
MAX
Term.
Utility
0
a
b
c
-1
0
0
+1
0
-1
0 -1 +1 0 -1 -1 0 +1 -1 0 +1 -1 -1 0 -1 0 -1 0

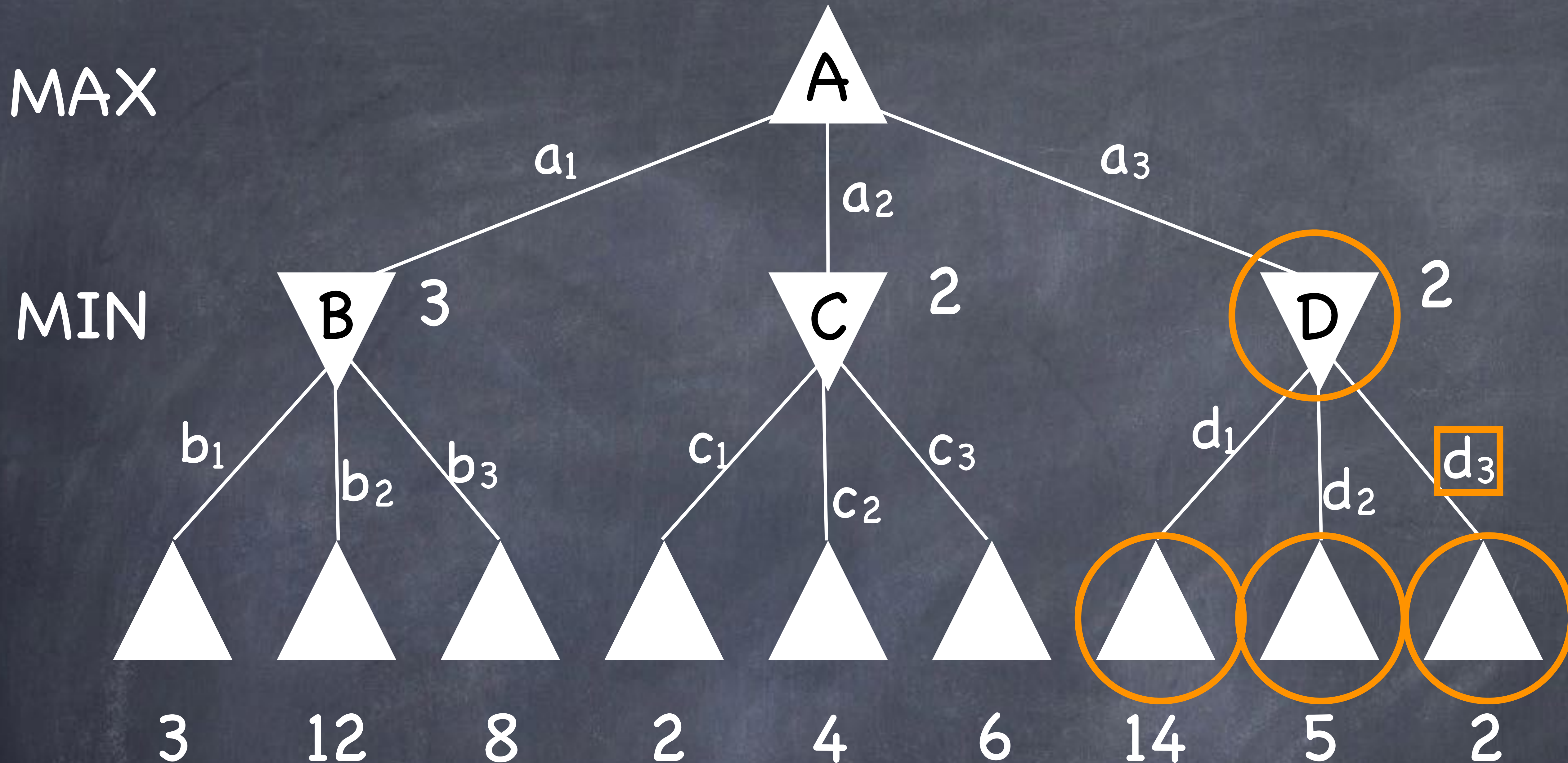
Minimax Algorithm

$$
\textsc{Minimax}(s) =
\begin{cases}
\textsc{Utility}(s) & \text{if } \textsc{Terminal-Test}(s) \\
\max_{a \in \textsc{Actions}(s)} \textsc{Minimax}(\textsc{Result}(s, a)) & \text{if } \textsc{Player}(s) = \textsc{max} \\
\min_{a \in \textsc{Actions}(s)} \textsc{Minimax}(\textsc{Result}(s, a)) & \text{if } \textsc{Player}(s) = \textsc{min}
\end{cases}
$$

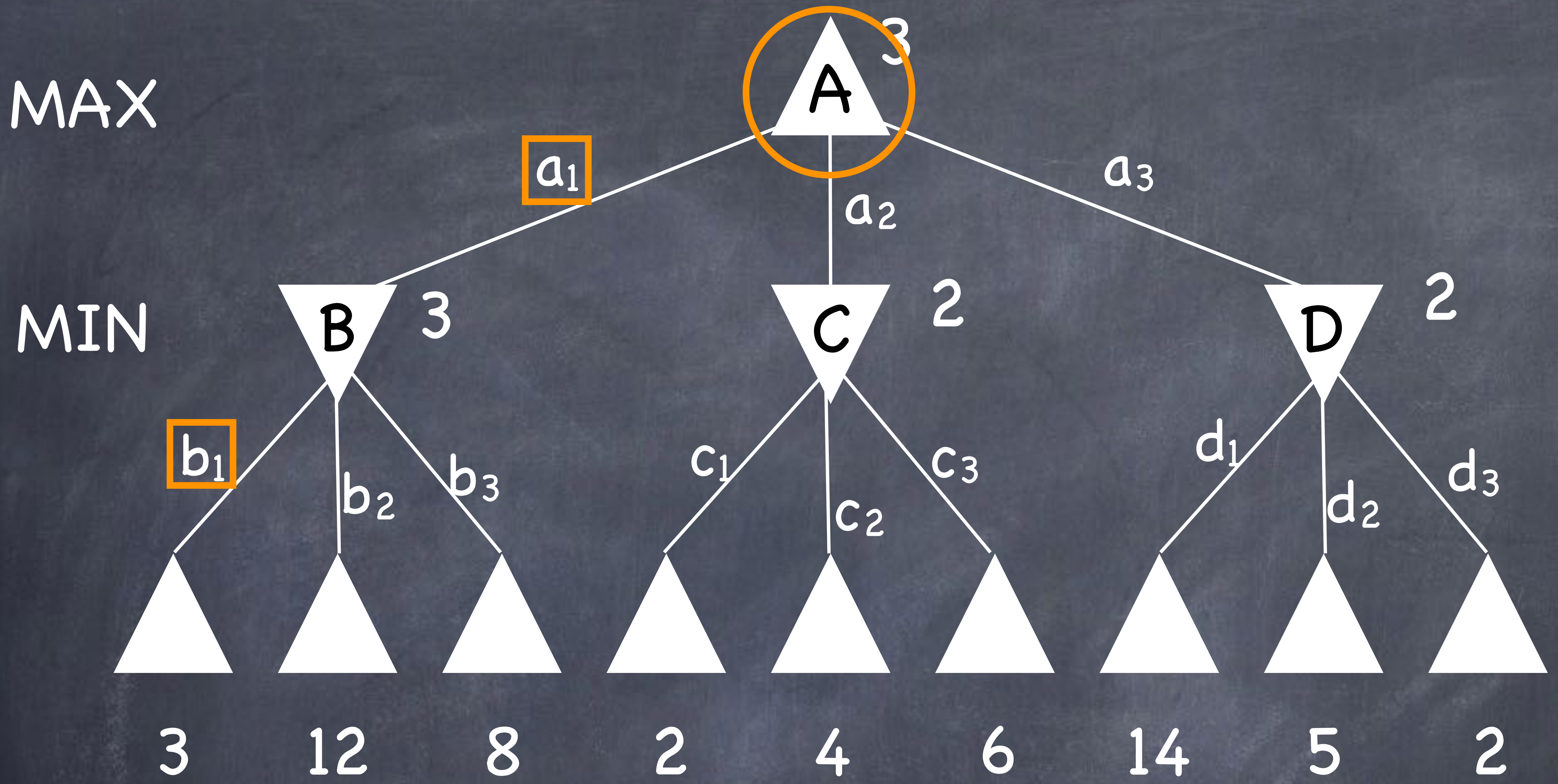

MAX
MIN
A
a_1
a_2
a_3
B
3
C
D
b_1
b_2
b_3
c_1
c_2
c_3
d_1
d_2
d_3
3
12
8



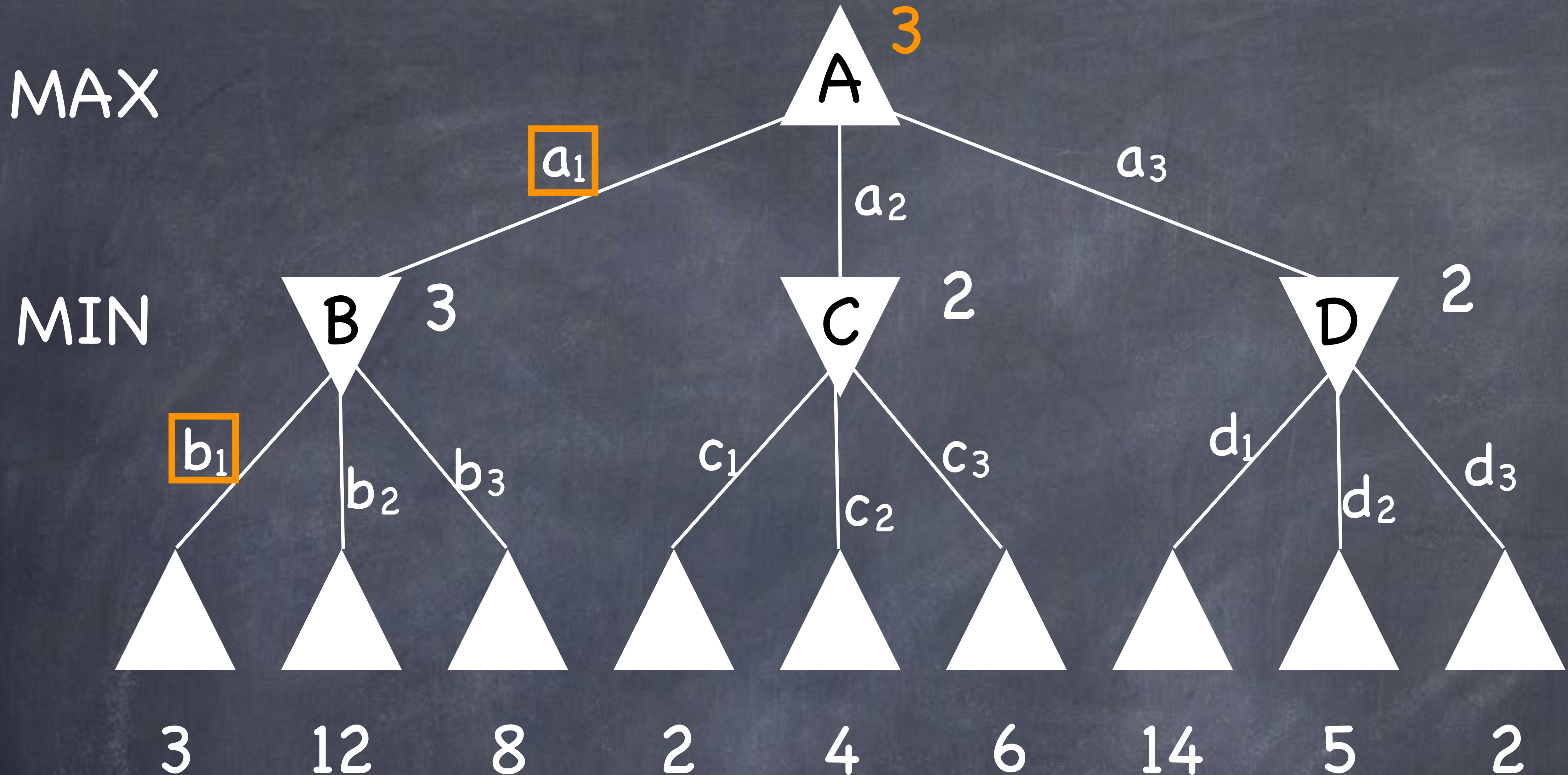
MAX
MIN
A
B
C
D
a_1
a_2
a_3
3
2
b_1
b_2
b_3
c_1
c_2
c_3
d_1
d_2
d_3
3
12
8
2
4
6

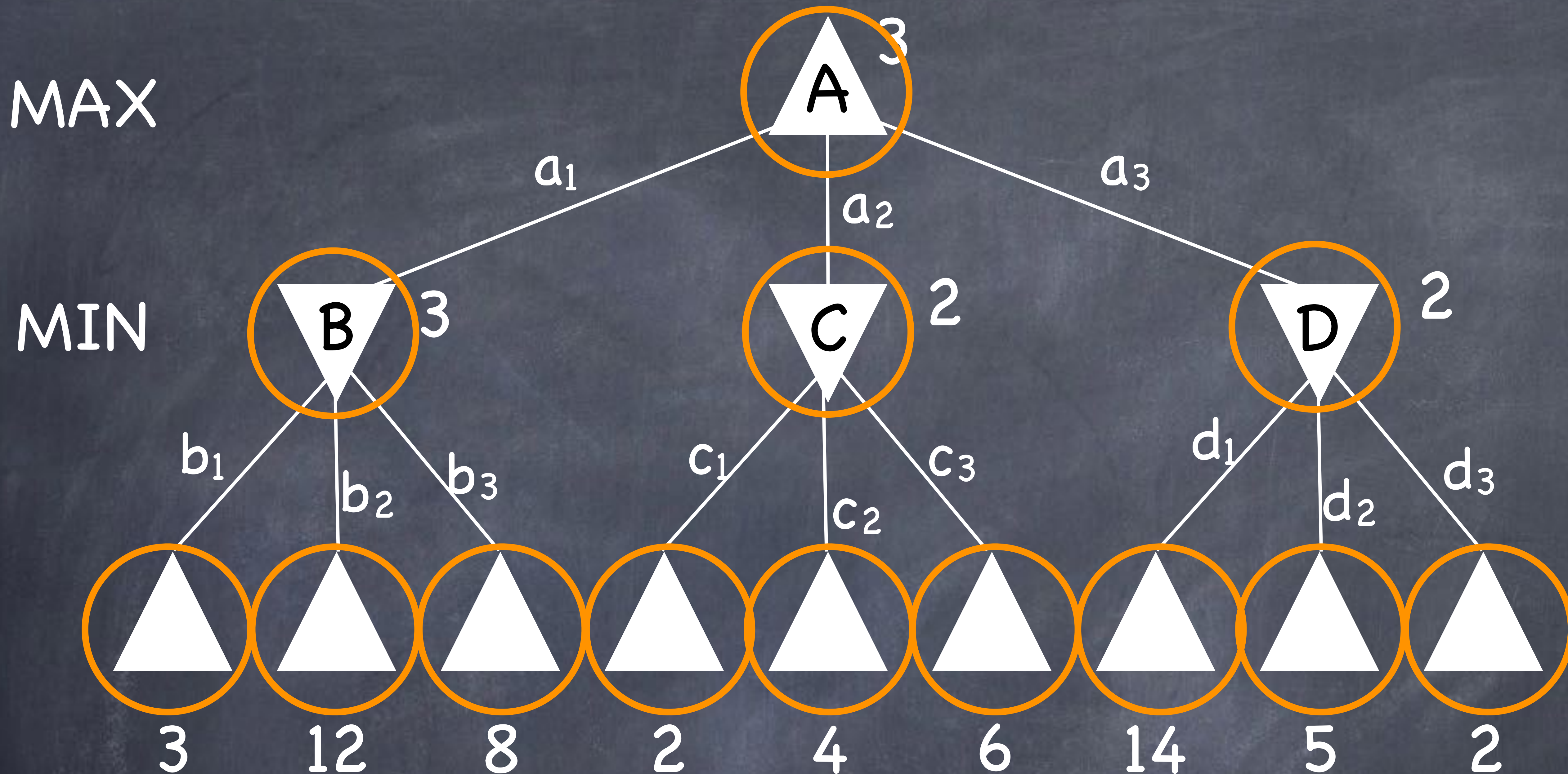


MAX
MIN
A
B
C
D
a_1
a_2
a_3
b_1
b_2
b_3
c_1
c_2
c_3
d_1
d_2
d_3
3
2
2
3
12
8
2
4
6
14
5
2



MAX
MIN
A
3
a1
a2
a3
B
3
C
2
D
2
b1
b2
b3
c1
c2
c3
d1
d2
d3
3
12
8
2
4
6
14
5
2


MAX
MIN
A
3
B
3
C
2
D
2
a1
a2
a3
b1
b2
b3
c1
c2
c3
d1
d2
d3
3
12
8
2
4
6
14
5
2

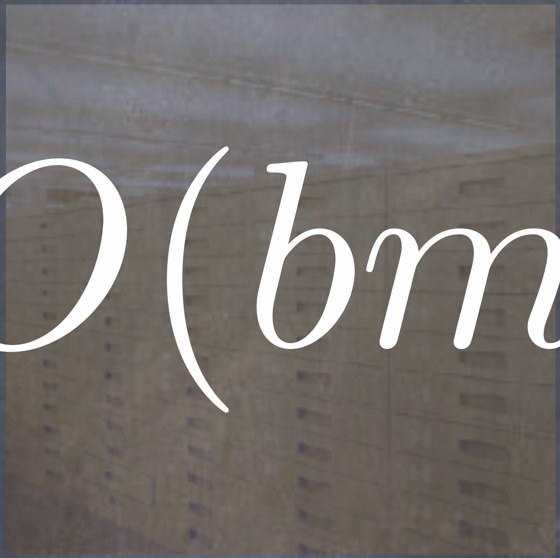


MAX
MIN
A
3
B
3
C
2
D
2
a_1
a_2
a_3
b_1
b_2
b_3
c_1
c_2
c_3
d_1
d_2
d_3
3
12
8
2
4
6
14
5
2

Minimax Analysis

$$O(b^m)$$

Time Complexity



$$O(bm)$$

Space Complexity

"for real games, the time cost is totally impractical"

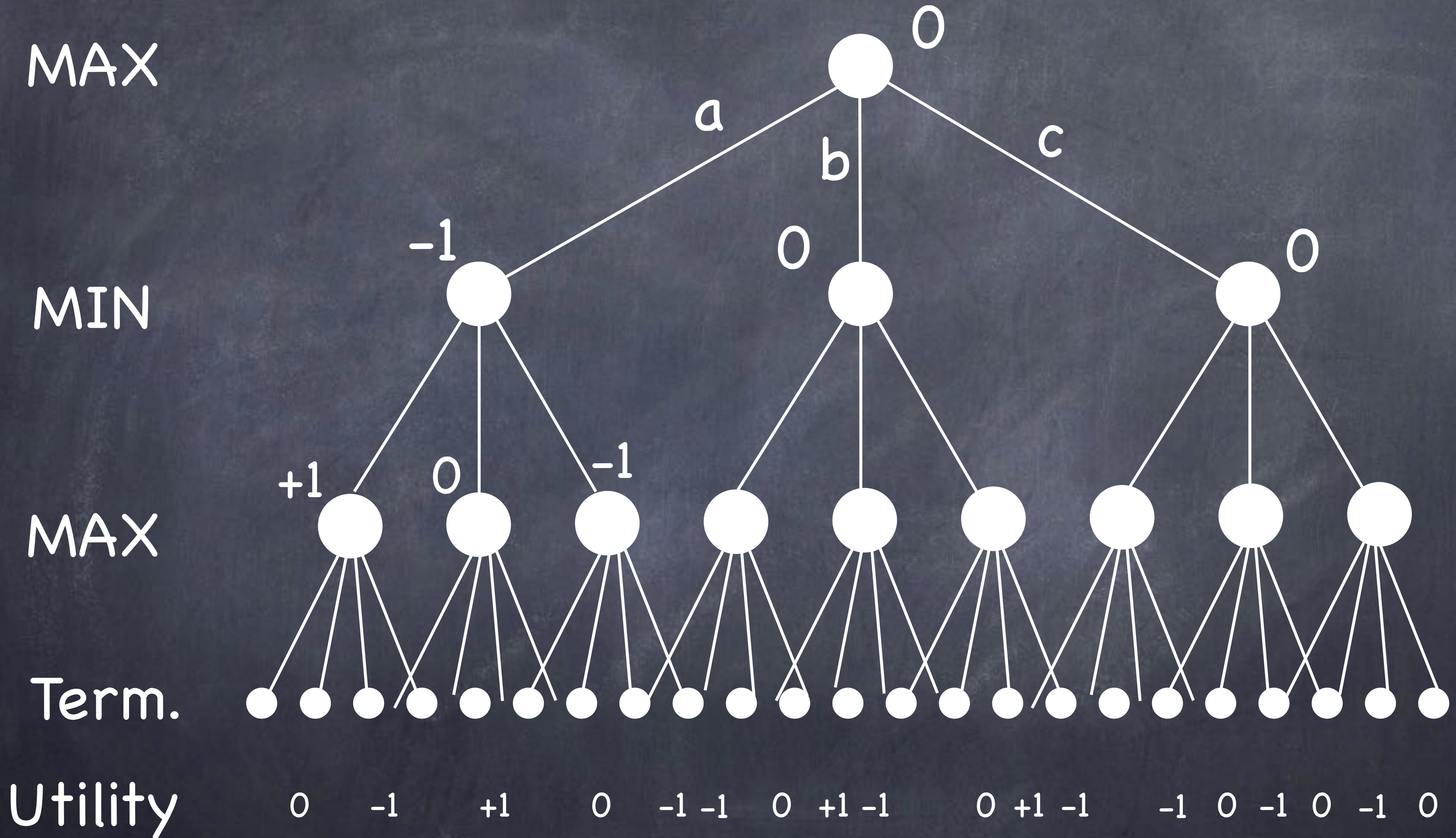
Minimax Summary

- Computes the optimal move assuming opponent also plays optimally (i.e., worst-case outcome)
- Explores game tree depth-first all the way to terminal states (end of game)
- Backs up utility values through alternating MIN and MAX (what's best for me is worst for you, and vice-versa)

Minimax Code

AIMA 5.2.1 and Figure 5.3 (page 166)

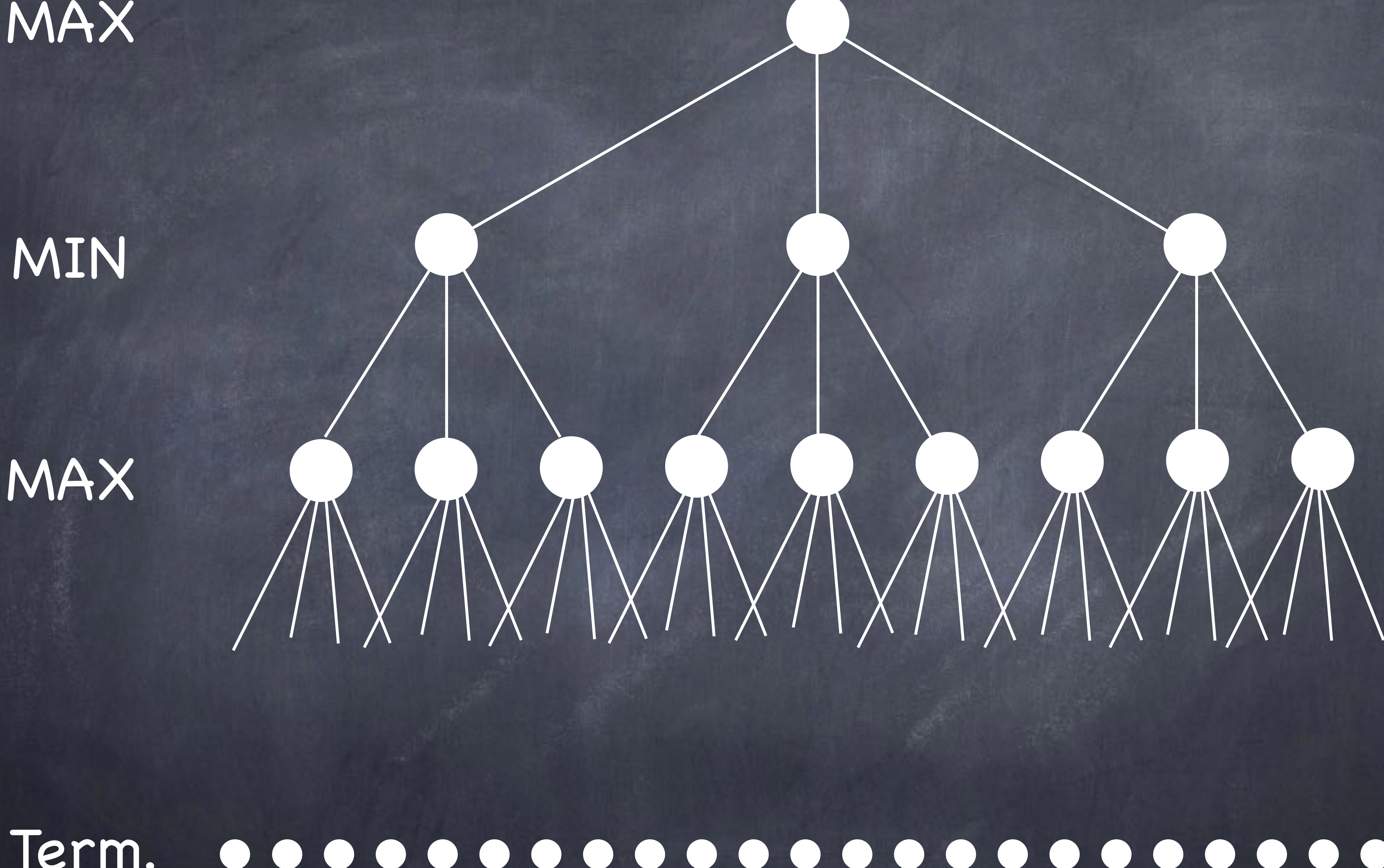
Not bad, but...

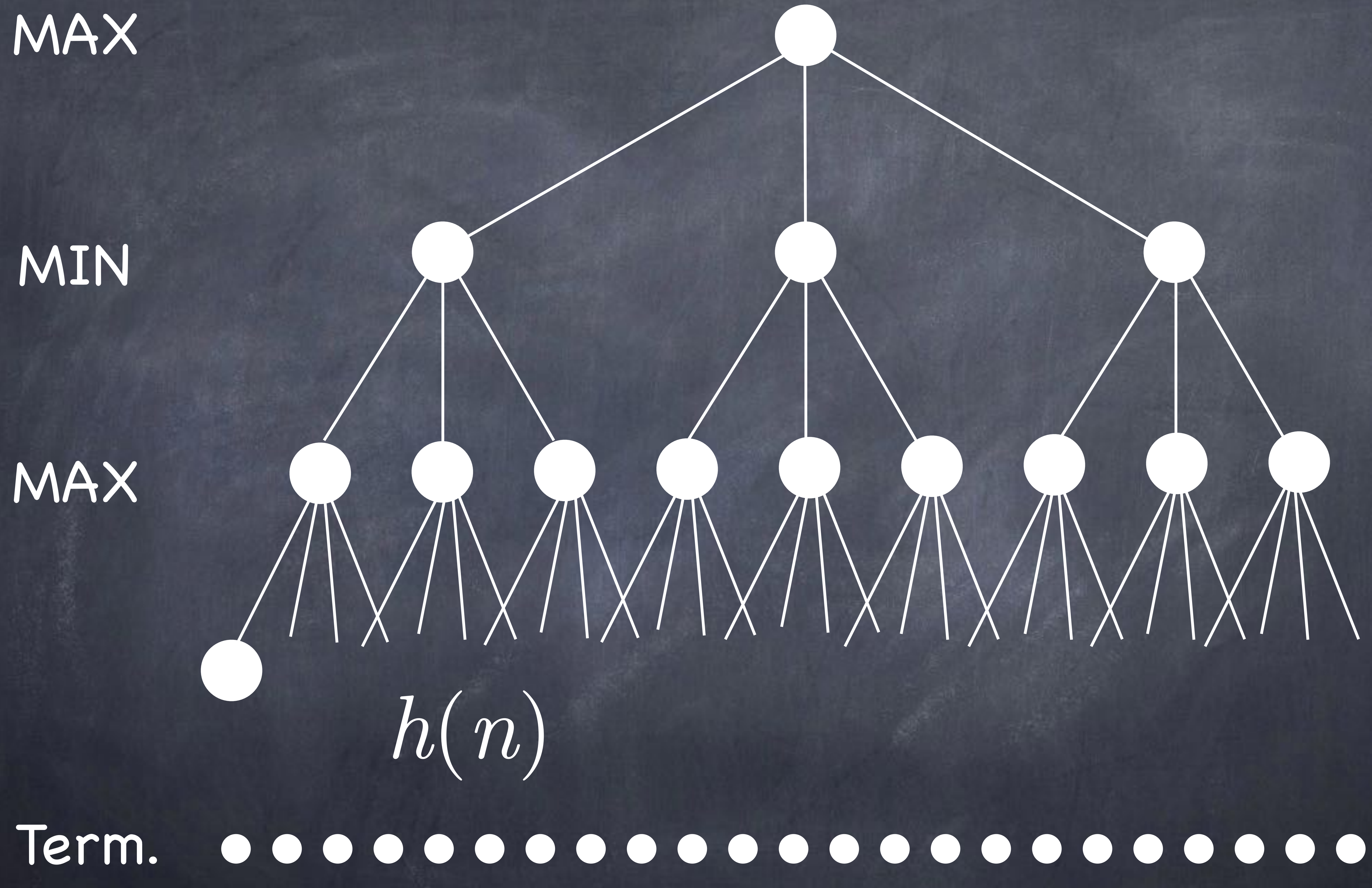

MAX
MIN
MAX
Term.
Utility
0
a
b
c
-1
0
0
+1
0
-1
0 -1 +1 0 -1 -1 0 +1 -1 0 +1 -1 -1 0 -1 0 -1 0

A Problem for Minimax

- You can't search all the way to the terminal nodes
- You can't evaluate a node (state) unless you're at a terminal node
- Utility function is defined on terminal states

Imperfect Real-Time Decisions


MAX
MIN
MAX
Term.


MAX
MIN
MAX
$h(n)$
Term.

Minimax Algorithm

$$\textsc{Minimax}(s) = \begin{cases} \textsc{Utility}(s) & \text{if } \textsc{Terminal-Test}(s) \\ \max_{a \in \textsc{Actions}(s)} \textsc{Minimax}(\textsc{Result}(s, a)) & \text{if } \textsc{Player}(s) = \textsc{max} \\ \min_{a \in \textsc{Actions}(s)} \textsc{Minimax}(\textsc{Result}(s, a)) & \text{if } \textsc{Player}(s) = \textsc{min} \end{cases}$$

Heuristic Minimax

$$\text{H-}\textsc{Minimax}(s) = \begin{cases} h(s) & \text{if } \textsc{Cutoff-Test}(s) \\ \max_{a \in \textsc{Actions}(s)} \textsc{Minimax}(\textsc{Result}(s,a)) & \text{if } \textsc{Player}(s) = \textsc{max} \\ \min_{a \in \textsc{Actions}(s)} \textsc{Minimax}(\textsc{Result}(s,a)) & \text{if } \textsc{Player}(s) = \textsc{min} \end{cases}$$

H-Minimax: Cutoff

- When to cutoff search?
  - Time
  - Depth
  - "Quiescence"
  - Can adjust dynamically
  - AIMA 5.4.2
- Combine with iterative deepening

H-Minimax: Heuristic

- How to evaluate non-terminal state?
  - AIMA 5.4.1
  - Chess example
    - Material value
    - Weighted sum
    - Strategic considerations
- Classic time-quality tradeoff

Heuristic Minimax

- Cutoff search before reaching terminal nodes (time, depth, “quiescence”)
- Use heuristic evaluation function to estimate state utility
- Backs up utility values through alternating MIN and MAX (what’s best for me is worst for you, and vice-versa)

Adversarial Search: MINIMAX

MINIMAX

- Searches to terminal nodes
- Uses utility function

H-MINIMAX

- Cuts off search before terminals
- Uses heuristic function

Backs up utility values through alternating MIN and MAX (zero-sum game)

For next time:

Chapter 5.3–5.4.2;
5.5–5.6; 5.7–5.9 fyi