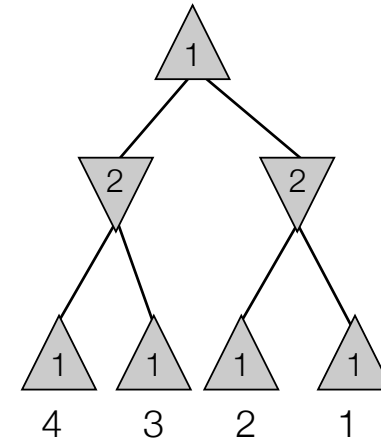


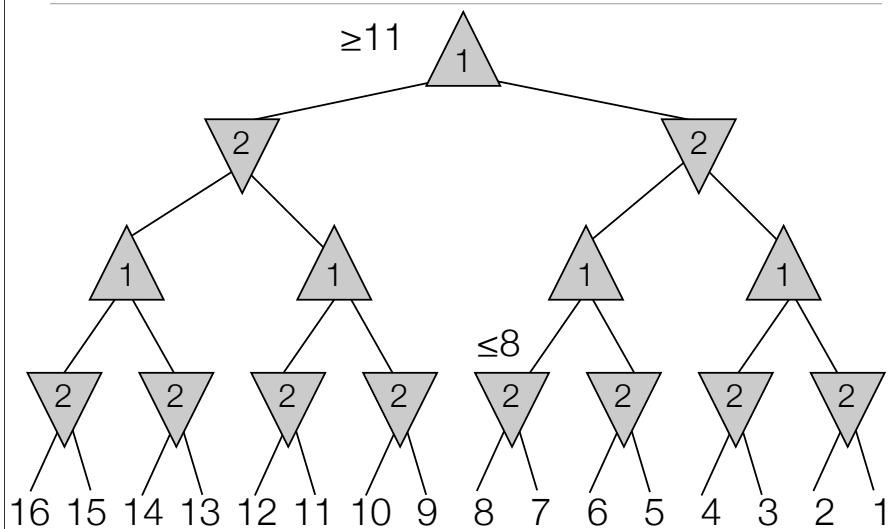
Lecture 5: Minimax, Alpha-Beta pruning

AI For Traditional Games
 Prof. Nathan Sturtevant
 Winter 2013

Alpha Beta Pruning (shallow)



Alpha Beta Pruning (deep)



Alpha-beta pruning

- Assuming the game is a perfectly ordered win/loss tree
 - Easily show that alpha-beta expands the min nodes
- Node ordering matters!
 - b^d with the wrong ordering
 - $b^{d/2}$ with perfect ordering
 - $b^{3d/4}$ with “average ordering”
- Will discuss move-ordering algorithms next week
- Is $b^{d/2}$ a big deal? Yes -- double our search depth

Analyze strategies

- What is needed to represent a full strategy for a single player?
 - Assume uniform branching factor b
 - Assume search tree depth d
 - Assume d is even
- This is an AND/OR tree
 - Need single choice at every MAX node
 - Need to consider all choices at every MIN node

(Strategy analysis cont'd)

- 1 move at root for MAX
- b moves at depth 2
- 1 move at depth 3 (MAX)
- $b^{\lfloor d/2 \rfloor}$ total leaf nodes
- Similar for MIN player
 - $b^{\lceil d/2 \rceil}$

Minimax strategy analysis

- Minimax shows that max can get at least p
- Minimax shows that min gets no less than p
- Requires a strategy for both players
- $b^{\lceil d/2 \rceil} + b^{\lfloor d/2 \rfloor} - 1 = O(b^{d/2})$
 - Strategies must overlap at a single leaf node
- Assumes that the game isn't a win at the root
 - Otherwise don't need the min strategy

Minimum tree example

- From Korf

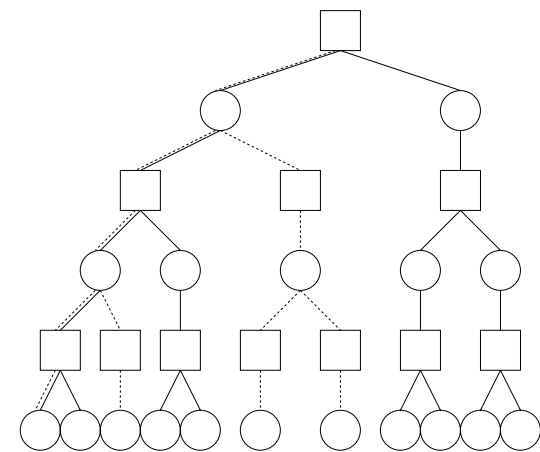


Figure 8.1: Minimal Depth-Five Binary Minimax Tree

What depth should I search?

- Cannot know proper search depth *a priori*
- Need a method to dynamically choose search depth
 - Possibilities?
- Iterative deepening approach
 - Search all depths!