

## Pseudocode [\[edit\]](#)

---

The pseudo-code for depth limited minimax with alpha-beta pruning is as follows:<sup>[12]</sup>

```
function alphabeta(node, depth,  $\alpha$ ,  $\beta$ , maximizingPlayer) is
  if depth = 0 or node is a terminal node then
    return the heuristic value of node
  if maximizingPlayer then
    value :=  $-\infty$ 
    for each child of node do
      value := max(value, alphabeta(child, depth - 1,  $\alpha$ ,  $\beta$ , FALSE))
       $\alpha$  := max( $\alpha$ , value)
      if  $\alpha \geq \beta$  then
        break (*  $\beta$  cut-off *)
    return value
  else
    value :=  $+\infty$ 
    for each child of node do
      value := min(value, alphabeta(child, depth - 1,  $\alpha$ ,  $\beta$ , TRUE))
       $\beta$  := min( $\beta$ , value)
      if  $\beta \leq \alpha$  then
        break (*  $\alpha$  cut-off *)
    return value
```

```
(* Initial call *)
alphabeta(origin, depth,  $-\infty$ ,  $+\infty$ , TRUE)
```

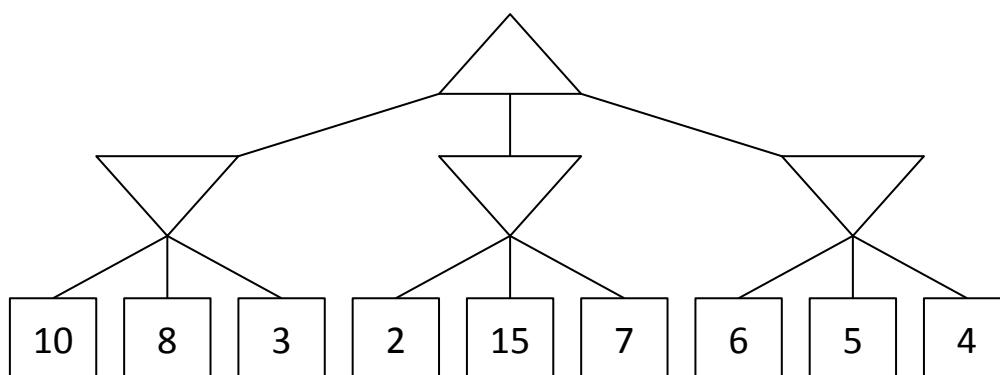
---

# Section 3

---

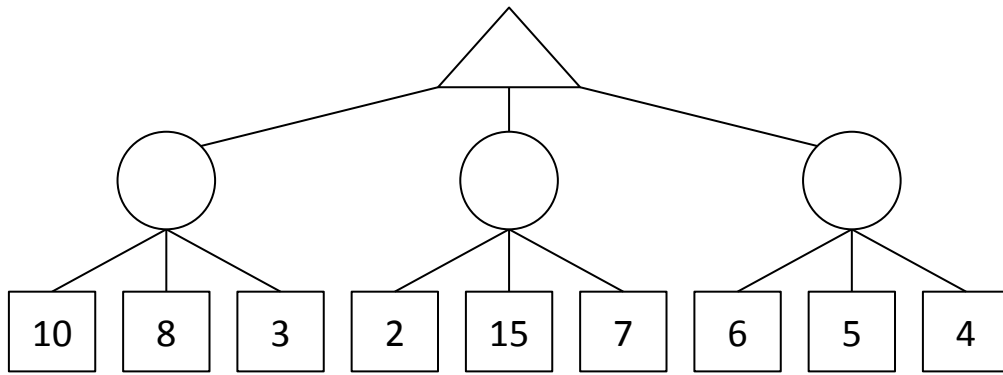
## 1 Games

- (a) Consider the zero-sum game tree shown below. Triangles that point up, such as at the top node (root), represent choices for the maximizing player; triangles that point down represent choices for the minimizing player. Assuming both players act optimally, fill in the minimax value of each node.



- (b) Which nodes can be pruned from the game tree above through alpha-beta pruning? If no nodes can be pruned, explain why not. Assume the search goes from left to right; when choosing which child to visit first, choose the left-most unvisited child.

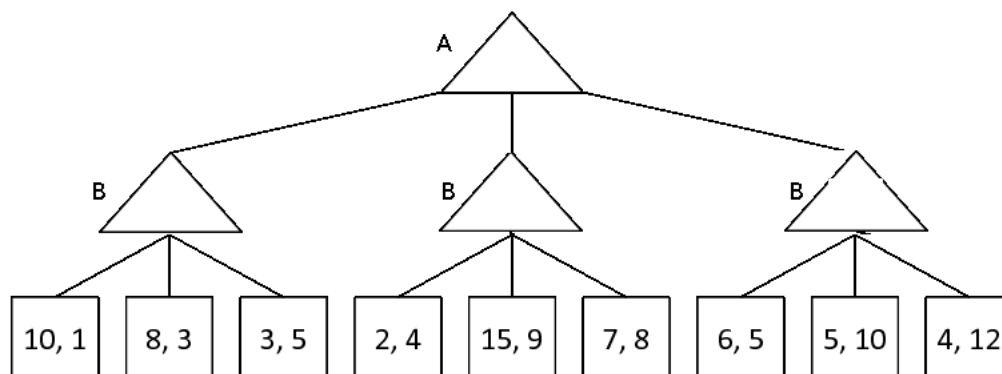
- (c) Again, consider the same zero-sum game tree, except that now, instead of a minimizing player, we have a chance node that will select one of the three values uniformly at random. Fill in the expectimax value of each node. The game tree is redrawn below for your convenience.



(d) Which nodes can be pruned from the game tree above through alpha-beta pruning? If no nodes can be pruned, explain why not.

## 2 Nonzero-sum Games

1. Let's look at a non-zero-sum version of a game. In this formulation, player A's utility will be represented as the first of the two leaf numbers, and player B's utility will be represented as the second of the two leaf numbers. Fill in this non-zero game tree assuming each player is acting optimally.



2. Which nodes can be pruned from the game tree above through alpha-beta pruning? If no nodes can be pruned, explain why not.