

```

1  import math
2
3  class Node:
4      def __init__(self, value=None):
5          self.value = value
6          self.children = []
7
8  def minimax(node, depth, maximizing_player):
9      if depth == 0 or not node.children:
10         return node.value
11
12     if maximizing_player:
13         max_eval = -math.inf
14         for child in node.children:
15             eval = minimax(child, depth - 1, False)
16             max_eval = max(max_eval, eval)
17         return max_eval
18     else:
19         min_eval = math.inf
20         for child in node.children:
21             eval = minimax(child, depth - 1, True)
22             min_eval = min(min_eval, eval)
23         return min_eval
24
25  def alpha_beta_pruning(node, depth, alpha, beta, maximizing_player):
26      if depth == 0 or not node.children:
27         return node.value
28
29     if maximizing_player:
30         max_eval = -math.inf
31         for child in node.children:
32             eval = alpha_beta_pruning(child, depth - 1, alpha, beta, False)
33             max_eval = max(max_eval, eval)
34             alpha = max(alpha, eval)
35             if beta <= alpha:
36                 break
37         return max_eval
38     else:
39         min_eval = math.inf
40         for child in node.children:
41             eval = alpha_beta_pruning(child, depth - 1, alpha, beta, True)
42             min_eval = min(min_eval, eval)
43             beta = min(beta, eval)
44             if beta <= alpha:
45                 break
46         return min_eval
47
48  # Example usage
49  if __name__ == "__main__":
50      root = Node()
51      root.children = [Node(3), Node(6), Node(8)]
52      root.children[0].children = [Node(4), Node(2)]
53      root.children[1].children = [Node(9), Node(1)]
54      root.children[2].children = [Node(5), Node(7)]
55
56      print("Minimax result:", minimax(root, 2, True))
57      print("Alpha-Beta Pruning result:", alpha_beta_pruning(root, 2, -math.inf, math.inf, True))
58

```

➡ Minimax result: 5
Alpha-Beta Pruning result: 5

