

## **ASSIGNMENT 10: ALPHA BETA PRUNING**

```
def alphabeta(node, depth, alpha, beta, maximizingPlayer, values, index=0):
    # Leaf node condition
    if depth == 0 or index >= len(values):
        return values[index]

    if maximizingPlayer:
        best = float('-inf')
        for i in range(2): # Two children for each node
            val = alphabeta(node*2+i, depth-1, alpha, beta, False, values, index*2+i)
            best = max(best, val)
            alpha = max(alpha, best)
            if beta <= alpha:
                break # Beta cut-off
        return best
    else:
        best = float('inf')
        for i in range(2):
            val = alphabeta(node*2+i, depth-1, alpha, beta, True, values, index*2+i)
            best = min(best, val)
            beta = min(beta, best)
            if beta <= alpha:
                break # Alpha cut-off
        return best

# Example: Game tree with depth = 3
values = [3, 5, 6, 9, 1, 2, 0, -1] # Leaf node values
depth = 3
result = alphabeta(0, depth, float('-inf'), float('inf'), True, values)

print("Optimal value (with Alpha-Beta Pruning):", result)
```