# Identifying Pruning Opportunities with Alpha-Beta

## **1. Game Tree Structure**

MAX  
 / \

MIN MIN /   /  
8 7 5 9

**Leaf nodes (utility values) left to right:** 8, 7, 5, 9

## **2. Alpha-Beta Tracing**

* **Initialize:**
  + Alpha (α) = -∞
  + Beta (β) = +∞

### **Step 1: Explore left MIN branch**

* MAX node α = -∞
* Left MIN node β = +∞

**Leaf 1:** 8  
- MIN chooses min(∞, 8) → β = 8  
- MAX α = max(-∞, β) → α = 8

**Leaf 2:** 7  
- MIN chooses min(β, 7) → β = 7  
- MAX α = max(-∞, β) → α = 7

**Left MIN node final value:** 7

### **Step 2: Explore right MIN branch**

* MAX node α = 7 (from left branch)
* Right MIN node β = +∞

**Leaf 3:** 5  
- MIN chooses min(β, 5) → β = 5

**Check for cutoff:**  
- Current β (MIN) = 5  
- MAX α = 7 (from left branch)  
- **Cutoff condition:** α ≥ β → 7 ≥ 5 ✅

* **Leaf 4 (value 9) is never evaluated**

**Right MIN node final value:** 5 (from evaluated leaf 5 only)

### **3. Root MAX Node Value**

* MAX chooses max of its MIN children: max(7, 5) → **7**

## **4. Pruning Explanation**

* The algorithm prunes **leaf node with value 9**.
* **Reason:**
  + At right MIN node:
    - Beta (β) = 5 after evaluating leaf 5
    - Alpha (α) at parent MAX = 7
  + Since α ≥ β (7 ≥ 5), MAX already has a better option from the left branch.
  + MIN cannot improve the outcome for MAX, so the remaining node (9) in this branch is **skipped**.

## **5. Summary**

| Node | Minimax Value | Evaluated? |
| --- | --- | --- |
| Left MIN | 7 | Yes |
| Right MIN | 5 | Yes (leaf 5), No (leaf 9) |
| Root MAX | 7 | N/A |

**Pruned Leaf Node:** 9  
**Final Root Value:** 7