**AND-OR Tree Evaluation**

Given complete binary tree with \( N = 2^k \) leaves, where levels are alternately AND/OR, and given values at the leaves, evaluate the value at root.

![Tree Diagram]

**Applications:**
- Game trees (minimax tree),
- Quantifier SAT
  \[
  \exists x_1 \forall x_2 \exists x_3 \ldots f(x_1, \ldots, x_k)
  \]

**Trivial Algorithm:** \( O(N) \) time

**Known Heuristics:** \( \alpha-\beta \) pruning, etc.

**Deterministic Lower Bound:** Every deterministic algorithm requires \( \Omega(N) \) time in worst case, in fact, has to inspect all \( N \) leaves.

**Proof Sketch:** by "adversary argument"

Say you are the algorithm, I = will construct a bad input (depending on the algorithm)

OR

AND
Snir's Alg'm ('85)

eval(u):
  if u is a leaf, return u's value
  let u₁, u₂ be u's children
  if u is OR
    if eval(u₁) = 1 return 1
    else return eval(u₂)
  if u is AND
    if eval(u₁) = 0 return 0
    else return eval(u₂)

(Las Vegas)

Analysis of Expected Time:

Call an OR node good if its value is 1, bad else
AND node good if its value is 0, bad else

let Gₖ = expected cost for a good node at level k
Bₖ = a " " " bad " " "

W.l.o.g. u is an OR node.

Case 1: u is bad.
  2 recursive calls
  \[ Bₖ \leq 2G_{k-1} \]

Case 2: u is good.
Case 2: \( u \) is good.

Subcase 1. \( u_1, u_2 \) both 1

1. recursive call

Subcase 2. \( u_1, u_2 \) one 1, one 0

Prob \( \frac{1}{2} \): \( \frac{1}{2} \) recursive call bad, \( \frac{1}{2} \) recursive call good

Prob \( \frac{1}{2} \): \( \frac{1}{2} \) recursive calls

\[ G_k \leq B_{k-1} + \frac{1}{2} G_{k-1} \]

Combine \( \Rightarrow \)

\[ G_k \leq 2 G_{k-2} + \frac{1}{2} G_{k-1} \]

"Fibonacci-like" recurrence

Guess \( G_k \leq x^k \).

Want \( 2x^{k-2} + \frac{1}{2} x^{k-1} = x^k \)

\[ 2 + \frac{1}{2} x = x^2 \]

\[ 2x^2 - x - 4 = 0 \]

\[ x = \frac{1 + \sqrt{33}}{4} \approx 1.69 \]

\( \Rightarrow \)

\[ G_{k-1} B_k = O\left( \left( \frac{1 + \sqrt{33}}{4} \right)^k \right) \quad k = \log_2 N \]

\[ \left( \frac{\log N}{N \log N} \right) \]

\[ = O\left( N^{\log_2 (1 + \sqrt{33})/4} \right) \]

\[ = O\left( N^{0.754} \right) \]

expected sublinear!

Rand. Lower Bd?

Yao's Principle: To prove lower bd on expected runtime of rand. Las Vegas algns on worst-case input.
Suffices to prove lower bound on 
expected runtime of det. algns 
on rand. input. 
(under an input distribution of your choices)

**Formally:** Fix input size n.

Let $A_1, A_2, \ldots$ be all correct det. algns.

Let $I_1, I_2, \ldots$ be all possible inputs.

Let $t_{ij} = \text{runtime of } A_i \text{ on } I_j$.

Rand. algm can be viewed as picking 
a random $A_i$ under distribution $p_i$.

Let $q_j = \text{prob. of picking } I_j$.

**Yao's Principle** (restated formally)

\[
\forall p_1, p_2, \ldots, \text{ with } \sum p_i = 1, \quad \forall q_1, q_2, \ldots \text{ with } \sum q_j = 1
\]

\[
\max_j \left( \sum_i p_i t_{ij} \right) \geq \min_i \left( \sum_j q_j t_{ij} \right)
\]

\[
\text{expected runtime} \quad \text{of rand. algm on } I_j
\]

\[
\text{expected runtime} \quad \text{of } A_i \text{ on rand. input}
\]

**Pf:**

\[
\max_j \left( \sum_i p_i t_{ij} \right) \geq \sum_j q_j \left( \sum_i p_i t_{ij} \right)
\]

\[
= \sum_{i,j} p_i q_i t_{ij}
\]

\[
= \sum_i p_i \left( \sum_j q_j t_{ij} \right)
\]

\[
\geq \min_j \sum_i q_j t_{ij}. \quad \square
\]

**Remark:** Also follows from LP duality
or von Neumann's minimax theorem
from classical 2-player game theory.
or von Neumann's minimax... 
from classical 2-player game theory.