"Average root → leaf length in a binary tree"

- have an arbitrary binary tree $T$ with leaf set $L(T)$.
- interested in the average root → leaf length, defined as

$$\text{AvgRL}(T) = \frac{1}{|L(T)|} \sum_{L \in L(T)} d(r, L),$$

where $d(r, L)$ is the number of edges in the path from the root $r$ of $T$ and the particular leaf $L$.

- Our goal is to lower bound $\text{AvgRL}(T)$ in terms of $\log(|L(T)|)$.

"Nice" binary trees, called "near complete", are binary trees where every internal node has two child nodes and every leaf is either at depth $h$ or depth $h-1$ ($h$ being the height of $T$).

Thm 11 (desired result for "near complete" trees)

If $T$ is "near complete" and has at least 4 leaves, then

$$\text{AvgRL}(T) \geq \frac{\log(|L(T)|)}{2}$$

Proof:

Let $h$ be the height of the tree $T$. For every leaf $L$, using the 'leaf' property of "near complete" trees gives

$$\text{AvgRL}(T) \geq (h-1) \quad \star$$

Also (using the two-child property for internal nodes) we have

$$2^{h-1} < |L(T)| \leq 2^h$$

hence $\log(|L(T)|) \leq h$, hence

$$\text{AvgRL}(T) \geq \log(|L(T)|) - 1 \geq \frac{\log(|L(T)|)}{2} \quad \text{for } |L(T)| \geq 4$$
Now have the desired lower bound on $\text{AvgRL}(T)$ for "near complete" trees — want a similar lower bound for all binary trees.

**Thm 12**

For any binary tree $T$, we can make a "near complete" tree $T'$ with the same number of leaves such that $\text{AvgRL}(T) \geq \text{AvgRL}(T')$ and we already know $\text{AvgRL}(T') \geq \log |L(T')|/2$ by Thm 11.

**proof:**

**STEP 1** Take $T$ and **contract** any internal nodes with only one child, to get an intermediate tree $T''$ where every internal node in $T''$ has two child nodes (this is one of the "near-complete properties" and $L(T') = L(T)$.

Note $d_{T''}(r, l) \leq d_T(r, l)$ for any leaf $l$, because contractions only decrease distance. So $\text{AvgRL}(T'') \leq \text{AvgRL}(T)$.

**STEP 2**

Take $T''$ and perform a series of pruning-and-reattaching moves on "low-lying" leaf siblings until all leaves are at depth $h$ or $(h-1)$ (for the appropriate height $h$)

Suppose $x$ is a "lowest leaf" and $z$ is a "highest leaf".

If $d(r, x) - d(r, z) \leq 1$, we have a "near complete" tree. **(DONE)**

We take the current tree to be $T'$.

If $d(r, x) - d(r, z) > 1$, not a "near complete" tree. **(NOT DONE)**

See details on next page of how we will "prune and reattach" to improve the height imbalance.
continuing proof of Thm 12

(Working on STEP 2)

x is a "lowest leaf" \(d(r, x)
\)
z is a "highest leaf" \(d(r, z)
\)
Difference \(d(r, x) - d(r, z) > 2\)

because we are not yet "near complete"

CURRENT TREE

by "2-child property" (STEP 1)
x must have a sibling
w must have two child nodes.
Also y has to be a leaf (by assumption x is "lowest leaf")

Result of "move"

I. \(L(T)\) is exactly the same.

II. All internal nodes have the same number (2)
of child nodes. (maintain the 2-child property)

III. Change in \(d(r, l)\) [leaf depth] values are:

\[\begin{align*}
\text{for } x \text{ and } y & : d(r, z) + 1 - d(r, x) \\
\text{for } z & : d(r, x) - 1 - d(r, z)
\end{align*}\]

Total change is \(d(r, z) + 1 - d(r, x) < 0\)

Avg RL has decreased (by \(d(r, x) > d(r, z) + 2\))

our "move"

chop off \(w\) and its two child-leaves x and y, and swap this "twig" with the leaf z.
implies that \[ \sum_{l \in L(T)} d(r, l) \]
decreases (by "at least 1") for each "move".

\[ \Rightarrow \]
So we can only perform the "move" finitely many times before the tree becomes balanced (all leaves are at depth \( h-1 \) or \( h \)).

\[ \Rightarrow \]
Let \( T' \) be the tree after it becomes balanced; the "near complete" tree.

Then \( \text{Avg}RL(T) > \text{Avg}RL(T') \)
(because STEP 1 (\( T \Rightarrow T' \)) only reduces \( \text{Avg}RL \),
and STEP 2 (\( T'' \Rightarrow T \) over a few moves) also reduces \( \text{Avg}RL \)).

We know \( \text{Avg}RL(T') \geq \frac{|L(T')|}{2} \) (for \( |L(T')| \geq 4 \))
by Thm 11.

\[ \Rightarrow \]
\( \text{Avg}RL(T) \geq \frac{|L(T)|}{2} \) (using \( L(T) = L(T') \))
for \( |L(T)| \geq 4 \).