Leftist Trees

* Linked Binary Tree
* Can do everything a heap does in the same time complexity.
* Can meld (merge) two leftist trees in $\mathcal{O}(\log n)$ time.

Extended Binary Tree

(Add external nodes)

Binary Tree

Extended Binary Tree
\textbf{S( ) Function}

\textit{S(x)}: For any node \( x \), \( S(x) \) is the length of a shortest path to an external node (in the subtree rooted at \( x \)).

\textbf{Computing \( S(x) \)}

\[
S(x) = \begin{cases} 
0 & \text{if } x \text{ is external node} \\
\min \{S(\text{lc}(x)), S(\text{rc}(x))\} + 1 & \text{o.w.} 
\end{cases}
\]
HEIGHT BIASED LEFTIST TREES (HBLT)

A BINARY TREE IS A HBLT IFF

for every internal node $x$

$s(lc(x)) \geq s(rc(x))$
PROPERTY 1

* A shortest root to external node path has length $s(\text{root})$. The rightmost path has this length.

PROPERTY 2

* The number of internal nodes is at least

$$2 \frac{s(\text{root})}{2} - 1$$

(level 1,...,$s(\text{root})$ do not have external nodes)

PROPERTY 3

* The rightmost path has length $O(\log n)$, $n$ is # of internal nodes.

PROP #2 $\Rightarrow n \geq 2 \frac{s(\text{root})}{2} - 1$

$\Rightarrow s(\text{root}) \leq \log (n+1)$

PROP #1 $\Rightarrow$ Rightmost path has length $s(\text{root})$
MIN HBLT

Insert Delete Min MELD()
INSERT  x = 7

DELETE MIN

meld

meld
MELD TWO BLTs

"TRAVERSE RIGHTMOST PATHS"
MELD

8

empty

But not HBLT
so swap children

Combine with previous
Swap not needed!
Combine with previous

Swap needed!
ARBITRARY DELETE

\[
A \rightarrow B \rightarrow L \rightarrow R
\]

--- delete

adjust S + leftist property

\[
A \rightarrow B \rightarrow L
\]

\[\Downarrow \text{now meld resulting HBLT+R}\]