Leftist Trees

- Linked binary trees.
- Insert and DeleteMin (or Delete Max) takes $O(\log n)$ time.
- Can Meld (Merge) two leftist trees in $O(\log n)$ time.
Extended Binary Trees

(Add external nodes)
**W( ) Weight Function**

\( W(x) \): For any node \( x \), \( W(x) \) is the total number of (internal) nodes in the subtree rooted at \( x \) (including \( x \)).
Computing $W(x)$

$$W(x) = \begin{cases} 
0 & \text{if } x \text{ is an external node} \\
W(lc(x)) + W(rc(x)) + 1 & \text{o.w.}
\end{cases}$$

where lc (rc) represents leftchild (rightchild).
Weight-Biased Leftist Trees (WBLT)

- A Binary tree is a WBLT
- iff
- for every internal node $x$,
  \[ W(lc(x)) \geq W(rc(x)) \]
Property of WBLTs

- A shorthest root to external node path has length $O(\log W(\text{Root}))$.
- The rightmost path has this length.
A Min WBLT that satisfies the “Min Heap ordering” is a Min WBLT.

The Insert, DeleteMin and Meld operations can be performed in $O(\log n)$ time.
Insert Operation

Insert $x$ in WBLT $H$ is just $\text{MELD}(x, H)$

Insert $x$ with value $8$ ⇒ Meld $H$ and the single node WBLT $x$ with value $8$
DeleteMin Operation

DeleteMin from WBLT $H$ is just $\text{MELD}(\text{lc}(H), \text{rc}(H))$

Delete Min $\Rightarrow$ Merge(lc(Root),rc(root))
Meld Two WBLTs

Traverse rightmost paths. See example beginning next page.
MELD TWO WBLT
PASTE BACK IN B

PASTE BACK IN A

Swap Lecture

Resulting WBLT