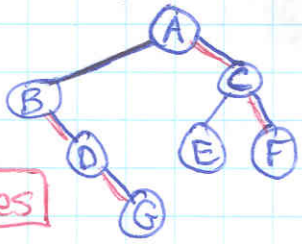
Better binary trie representations via transformations [Munro & Raman - SICOMP 2001]

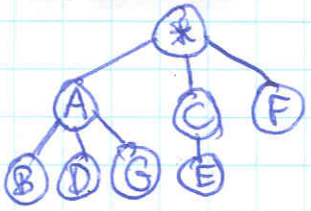
6.897
L 21.4
Apr. 26, 2005

"ordinal"

binary trie | rooted ordered tree | balanced parentheses (=bitstring)



n nodes



((()()())(()))
* A B B D D G G A C E E C F F *

2n bits (ignoring *'s)

node	node	left paren. (
left child	first child (if any)	next character (if ( )
right child	next sibling	char. after matching)
parent	{ previous sibling or parent	{ prev. char.) ⇒ matching (
		{ prev. char. (⇒ that (
subtree size	size(node) + sizes(right siblings)	½ · distance to enclosing)

- like (and using) rank & select,
  can find matching (/) & enclosing (/)
  in O(1) time using o(n) space [Munro & Raman - SICOMP 2001]
  ⇒ all these ops. in O(1) time & o(n) space
- from subtree size can accumulate index of node

Leaf operations: [Munro, Raman, Rao – J. Alg. 2001]

leaf	rightmost child of parent	()) ↑
leaf index/rank		$\text{rank}_{()}$ (here)
leaf size of subtree	leaf size(node) + leaf size(right sibl.)	$\text{rank}_{()}$ (matching) of parent(here)) − $\text{rank}_{()}$ (here)
leaf select		$\text{select}_{()}(i)$
leftmost leaf in subtree		leaf select (leaf index(here) + 1)
rightmost leaf in subtree		leaf select (leaf index (x)) where x = char. before matching) of parent(here)

Ordinal trees with ith child in O(1) [Benoit, Demaine, Munro, Raman, Raman, Rao – Alg.]