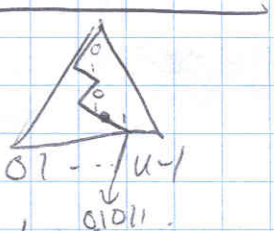
y-fast trees [Willard 1984]

6.897
L2.4
Feb. 12, 2003

- Write each elt as binary number = root-to-leaf path in tree.
- Store hash table of all prefixes of all elements.

$\Rightarrow O(n \lg u)$ space.



- In $O(\lg \lg u)$ time, can compute lowest marked ancestor of any universe element (binary search on length of match).
- By looking at min or max below that node find sucessor or predecessor.
- Find other via linked list of elements.

$\Rightarrow$ Successor & Pred. in $O(\lg \lg u)$ time
Insert/Delete: $O(\lg u)$.

Indirection: cluster elements into conseq. groups of size $\Theta(\lg u)$.

- within a group, use balanced BST to Insert/Delete/Succ./Pred in $O(\lg(\lg u))$ time.
- use hash table only on some representative elt. of each group.

$\Rightarrow O(n)$ space.

- search in hash table narrows down to 2 groups — look in both
- Insert/Delete: in group first.
  If a group grows by 2x or shrinks by 4x, split or merge (& then possibly split).
  This may change $O(1)$ representative elts
  $\Rightarrow O(\lg u)$ cost in hash table
  but only happens after $\Omega(\lg u)$ changes.
  $\Rightarrow O(1)$ amortized

$O(\lg \lg u)$ overall.