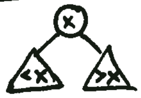
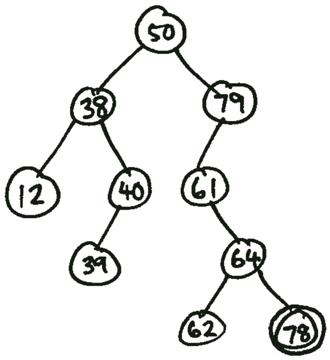
6.006 REVIEW SESSION — DECEMBER 11, 2008

TREES/BSTs





A. INSERT(78) $\Theta(h)$

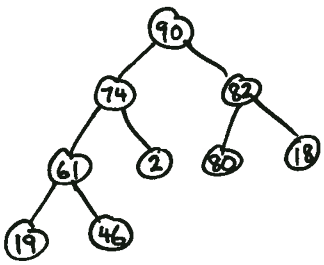
B. DELETE(50) — swap with 61 and delete $\Theta(h)$

C. SUCCESSOR $\Theta(h)$

D. IN-ORDER WALK $\Theta(n)$

– AND: CONSTRUCTING A BST FROM A SORTED LIST... (ALSO $\Theta(n)$)

HEAPS



NEARLY-COMPLETE BINARY TREE

← MAX-HEAP... OF COURSE, MIN-HEAPS EXIST, TOO.

B
~~A~~. EXTRACT-MIN* $\Theta(\lg n)$

*for min-heaps.

A
~~B~~. HEAPIFY $\Theta(\lg n)$

C. BUILD-HEAP $\Theta(n)$

D. PRIORITY QUEUE
- DECREASE-KEY*

ARRAY REPRESENTATION:

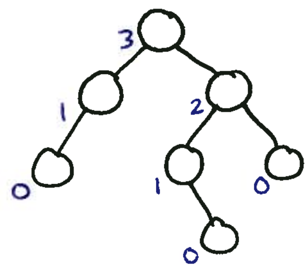
90	74	82	61	2	...
0	1	2	3	4	...

PARENT[j] = [redacted] $\lfloor \frac{j-1}{2} \rfloor$

CHILD[i]:

LEFT[i] = 2i + 1 RIGHT[i] = 2i+2

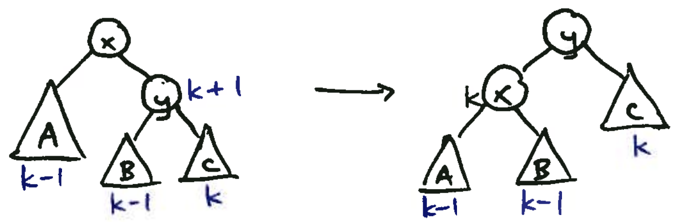
AVL TREES: BALANCED BSTs



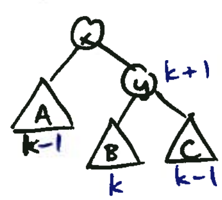
$1 \geq |h(\text{LEFT}) - h(\text{RIGHT})|$

REBALANCING ONLY REQUIRES ROTATIONS

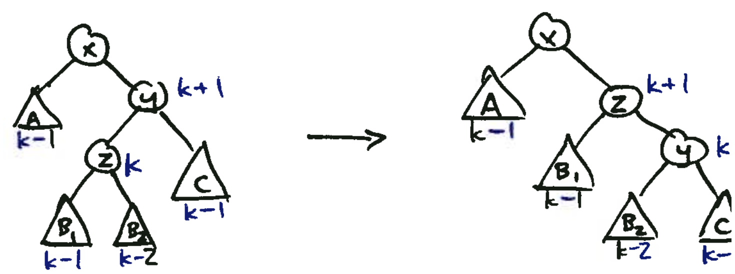
EASY CASE:



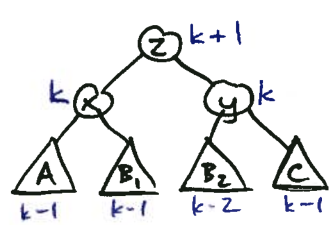
HARD CASE:



THE ABOVE WON'T WORK...
WE NEED TWO ROTATIONS

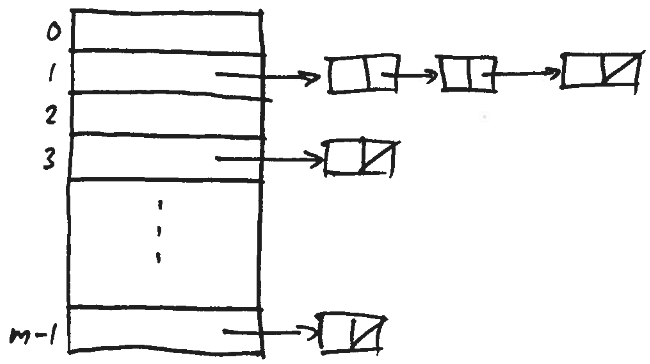


THEN IT BECOMES THE EASY CASE...



006 REVIEW SESSION — DECEMBER 11, 2008

HASHING / HASH TABLES



A. COLLISION RESOLUTION

B. INSERT

C. DELETE

D. LOOKUP

EXPECTED $\Theta(1)$

WORST CASE $\Theta(n)$

E. ROLLING HASHES

LOAD FACTOR $\alpha = \frac{n}{m}$

(n ELTS. INSERTED)

AMORTIZED ANALYSIS

HASH TABLE (IGNORE DELETIONS) — WANT TO KEEP $\alpha < 4/5$

ON INSERT, IF $\alpha \geq 4/5$, ALLOCATE A NEW TABLE OF $2m$ SIZE AND REHASH EVERYTHING

$\rightarrow$ $O(m+n) = O(n)$ TIME TO RESIZE

$O(1)$ OTHERWISE

IN A SERIES OF n INSERTIONS:

- suppose resizes occur at $k, 2k, 4k, \ldots$
- then cost of insertions is $O(n + (k + 2k + \ldots + n)) = O(n + (\frac{n}{2^i} + \frac{n}{2^{i-1}} + \ldots + n))$

$$= O(n + (2n))$$

$$= O(3n) = O(n)$$

Thus each insert is $\frac{O(n)}{n} = O(1)$ time