

# CS180 - Treaps

Note Title

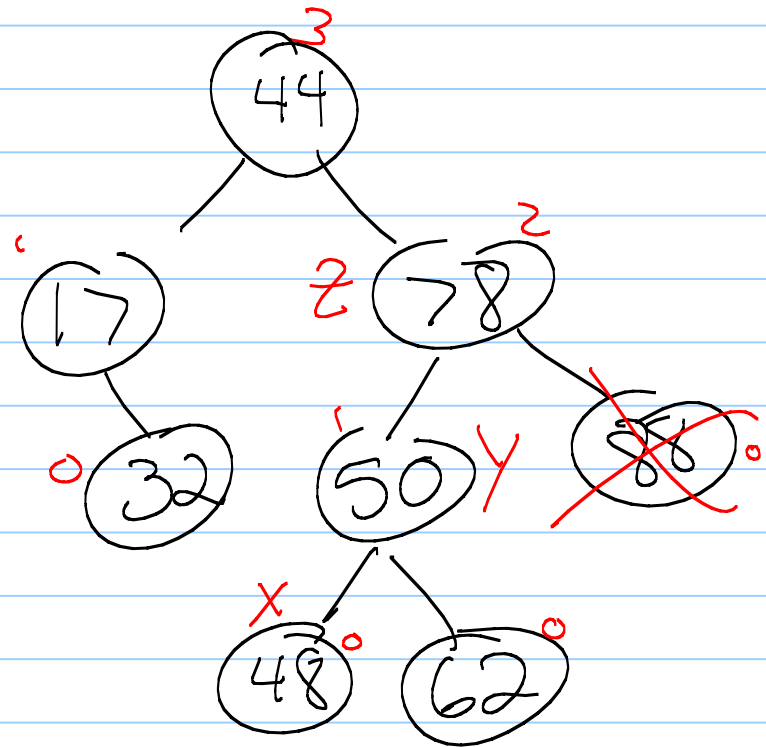
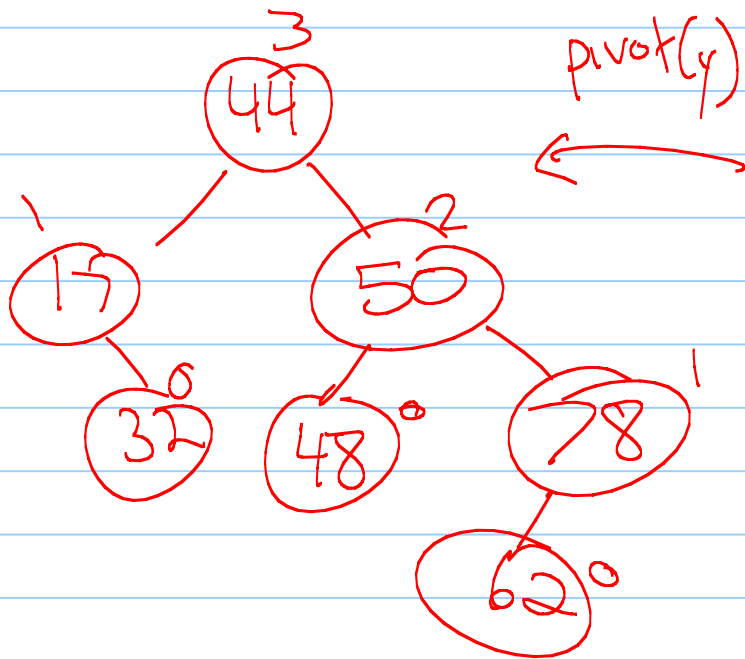
5/4/2011

## Announcements

- Next HW - due next Thursday
- Midterm 2 - a week from Monday  
(no lab next week)
- Next HW - after midterm

Next HW: remove in AVL

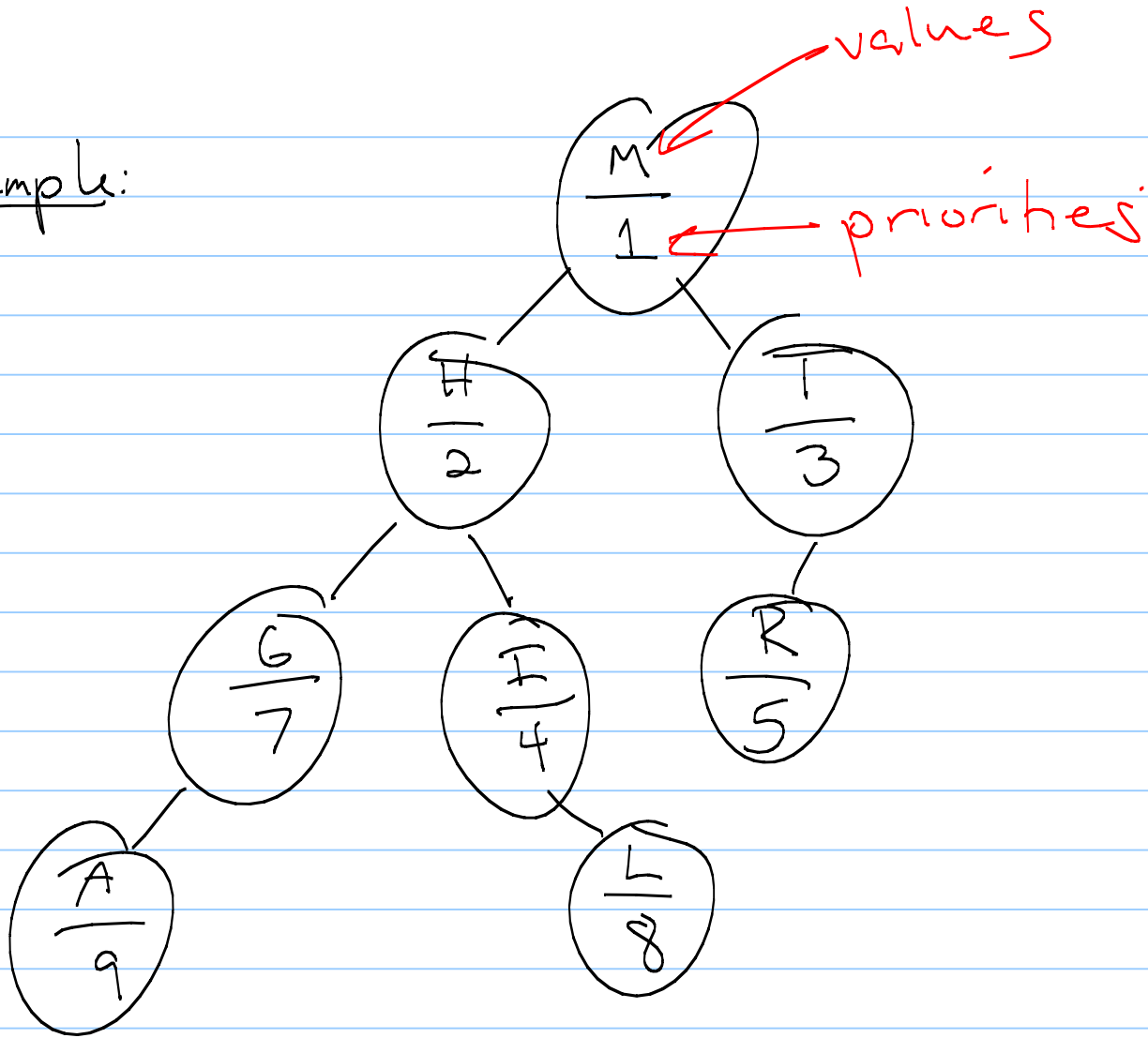
remove(88)



Treaps: a new binary tree data structure

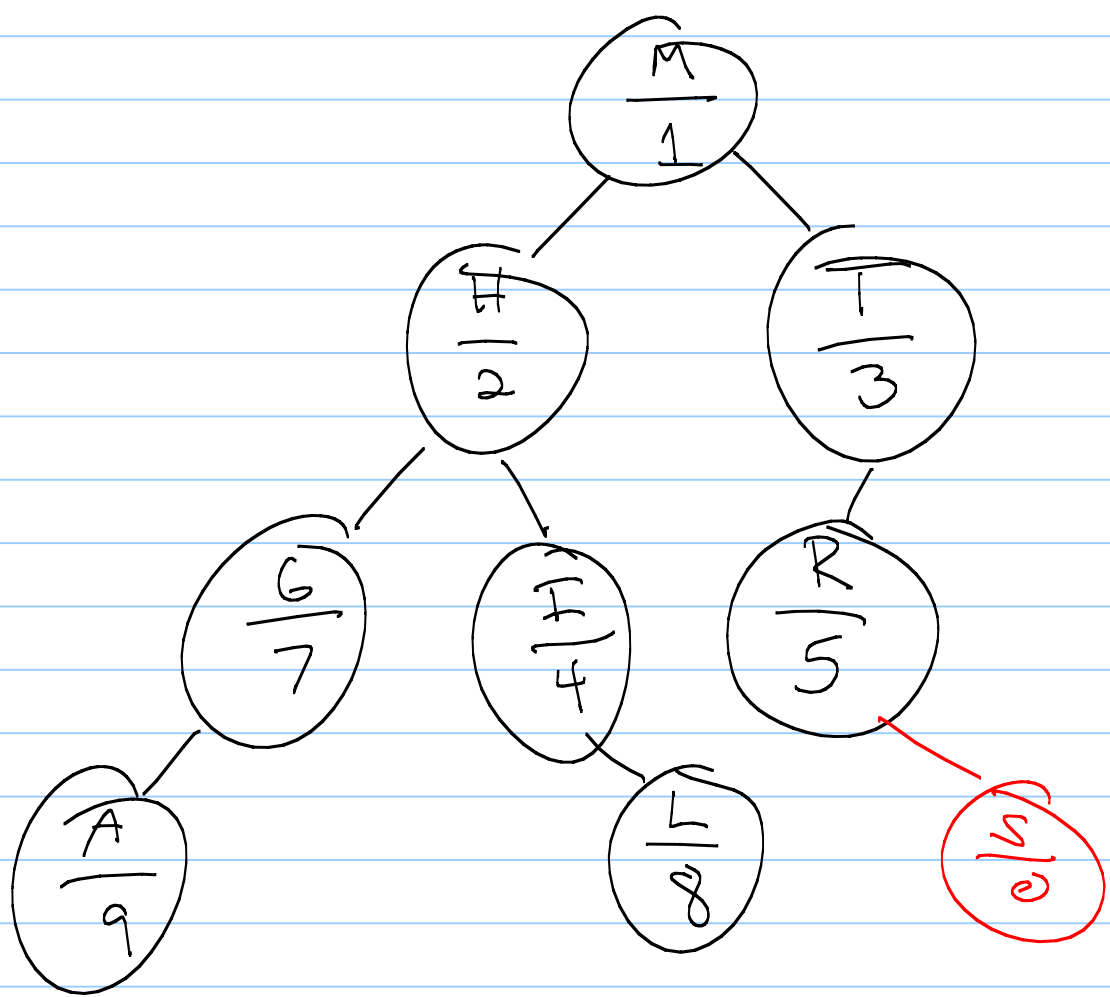
- Nodes will contain both values and priorities
- A treap is a BST over the values and a heap over the priorities.

Example:

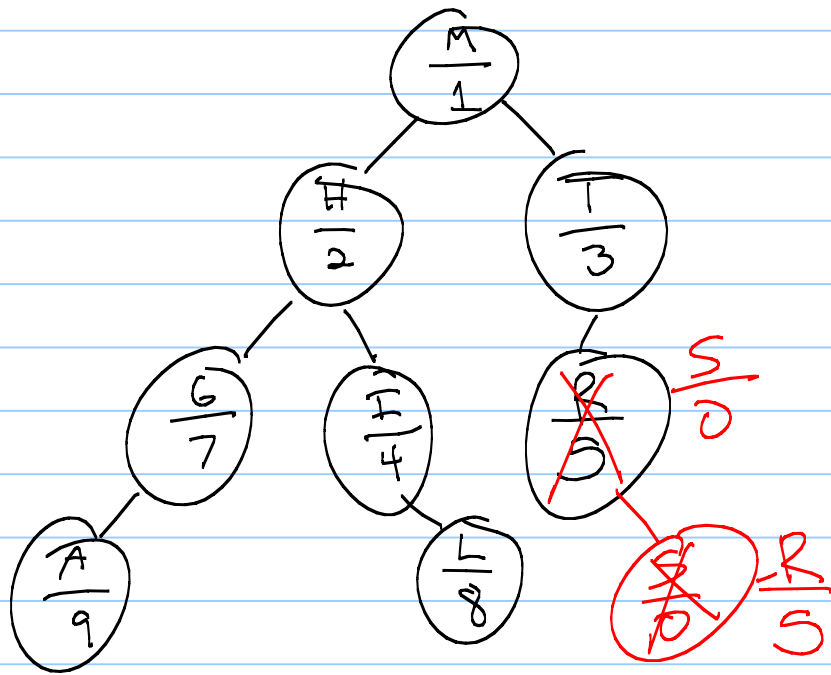


# Insert

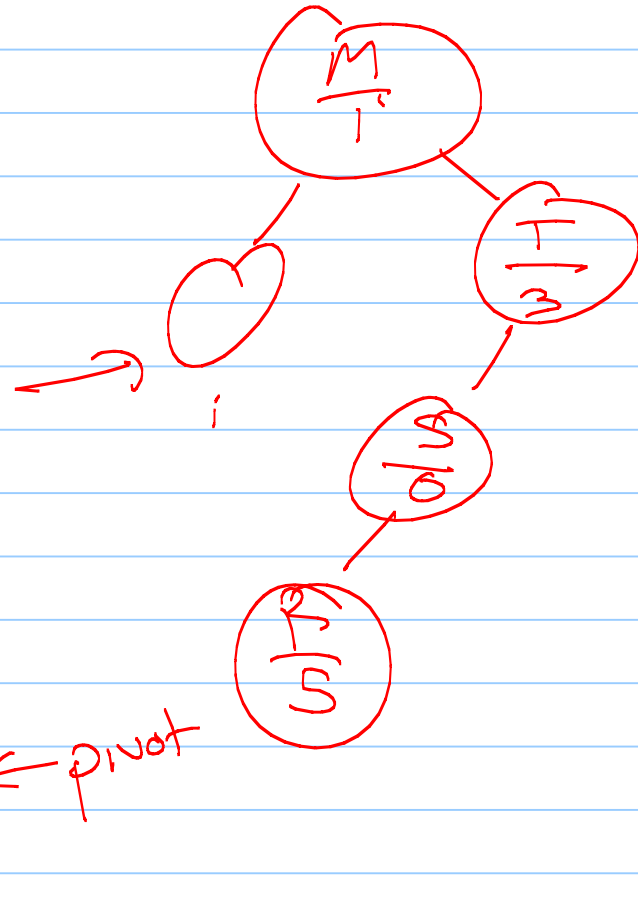
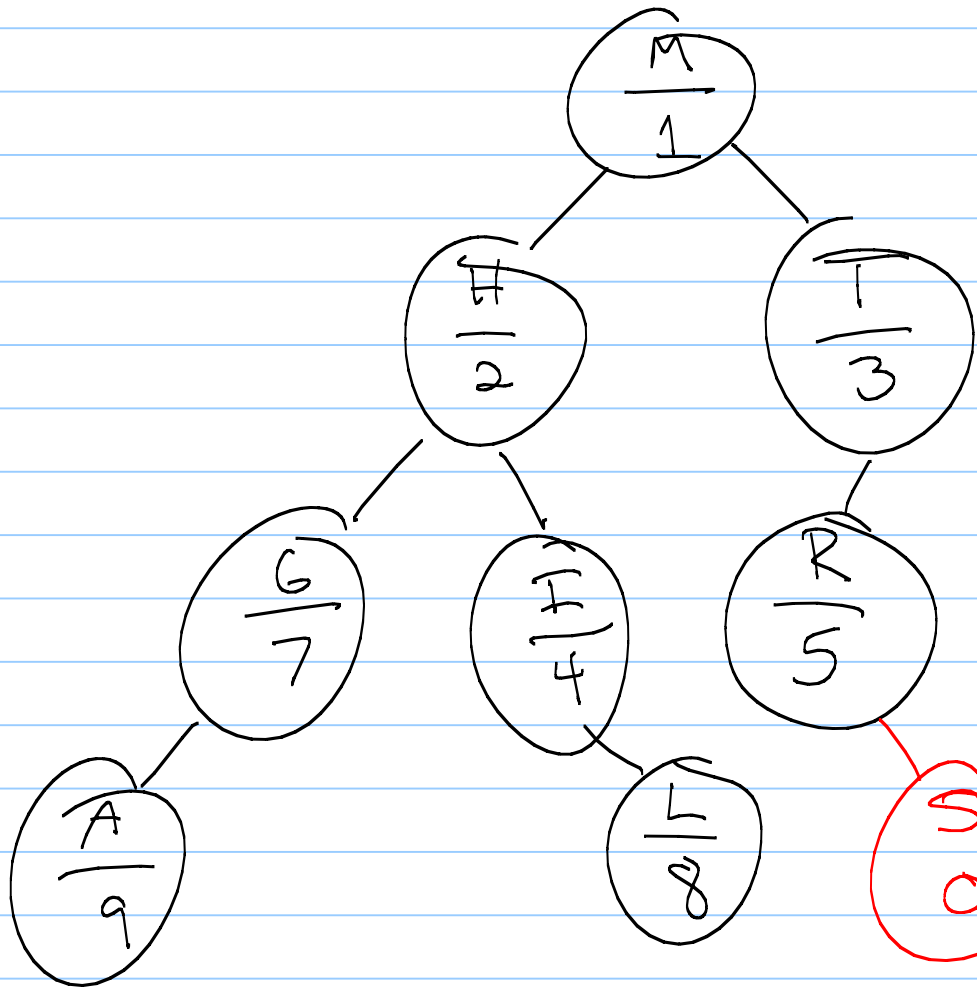
insert: (S, 0)

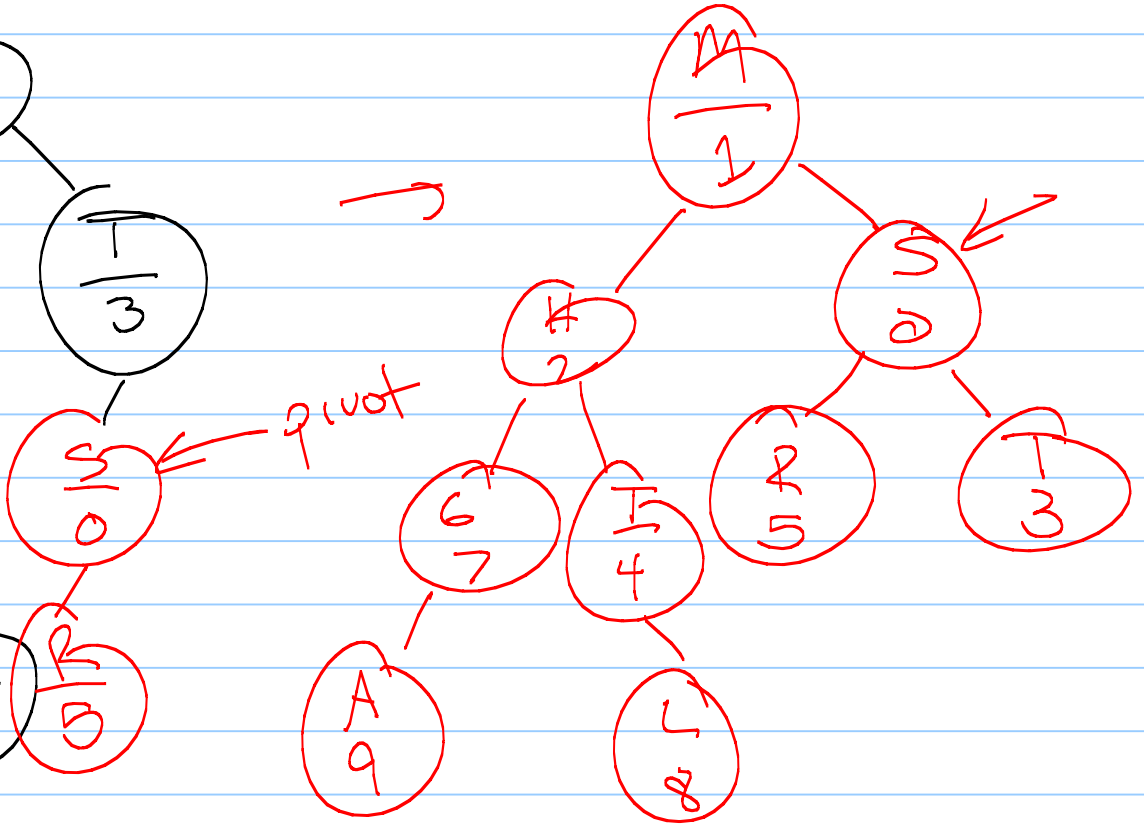
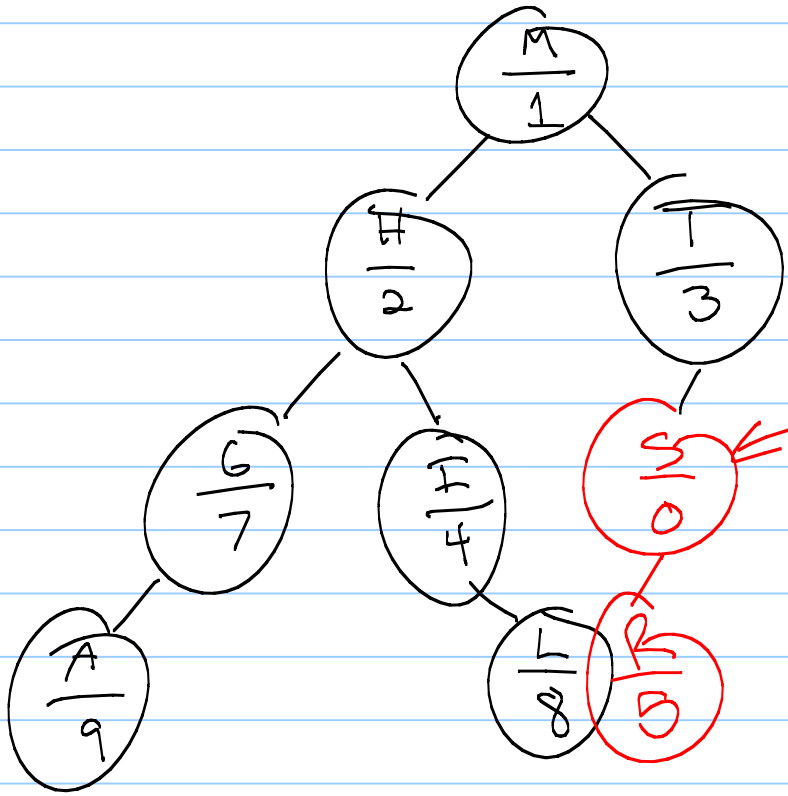


In heap, we "bubble up".  
Will that work here?

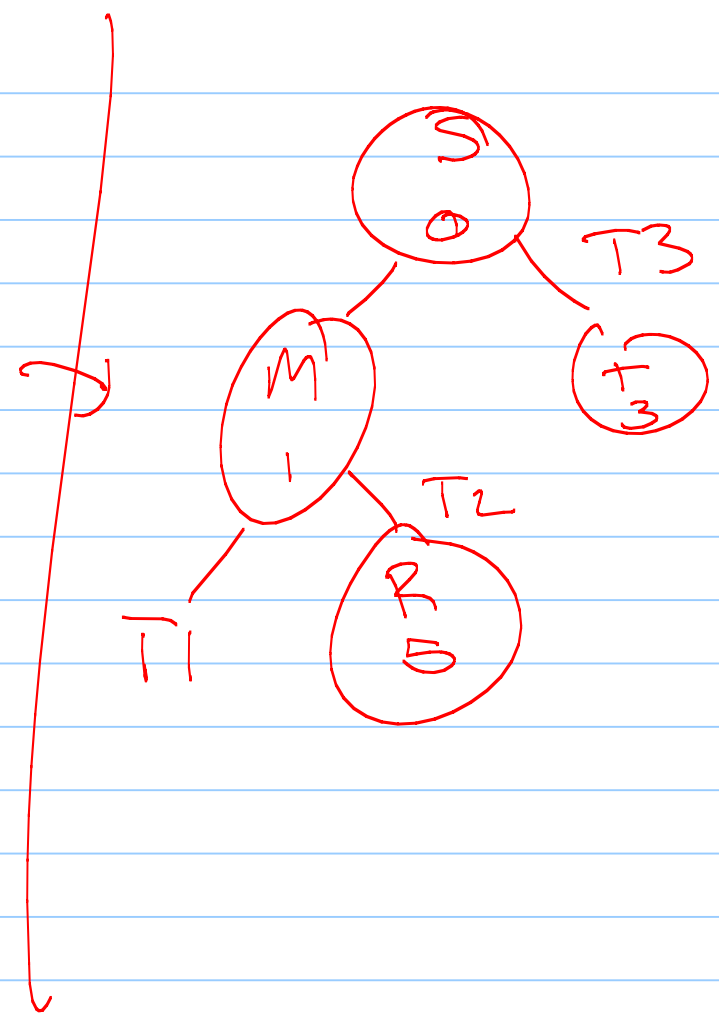
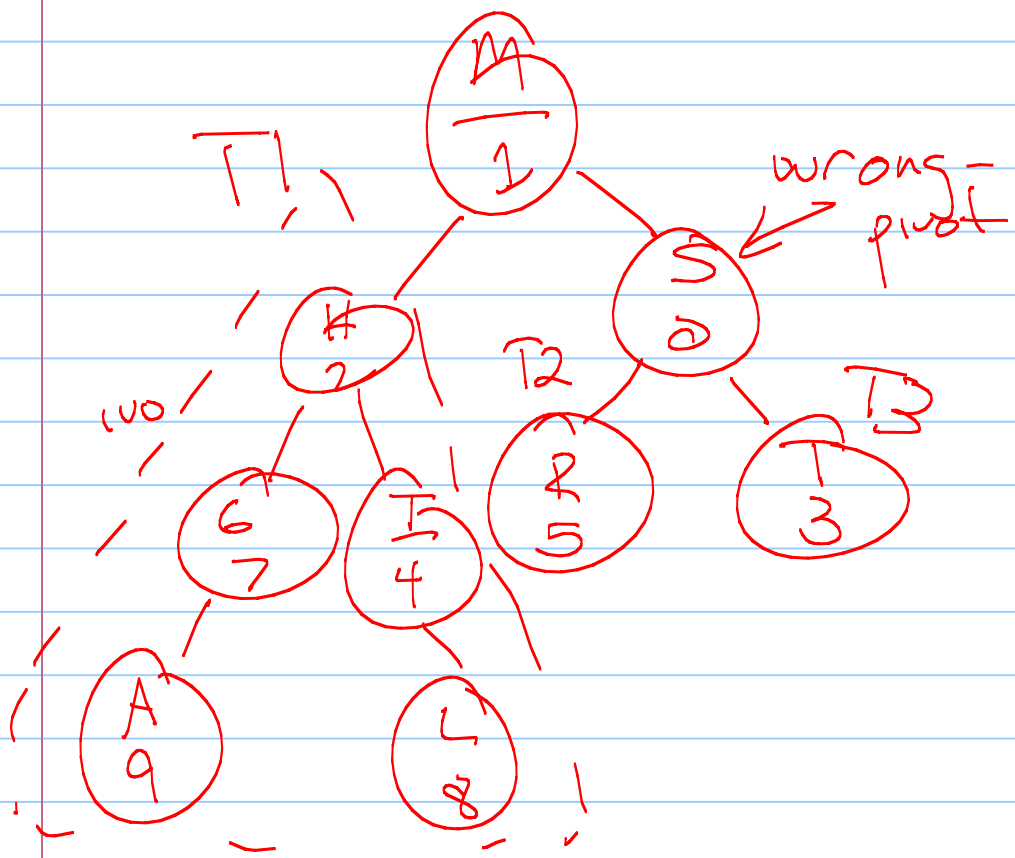


violates  
BST  
property  
pivot!



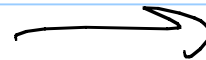
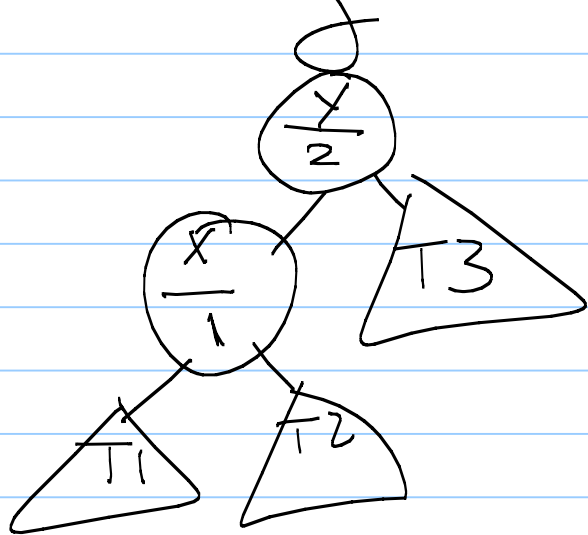




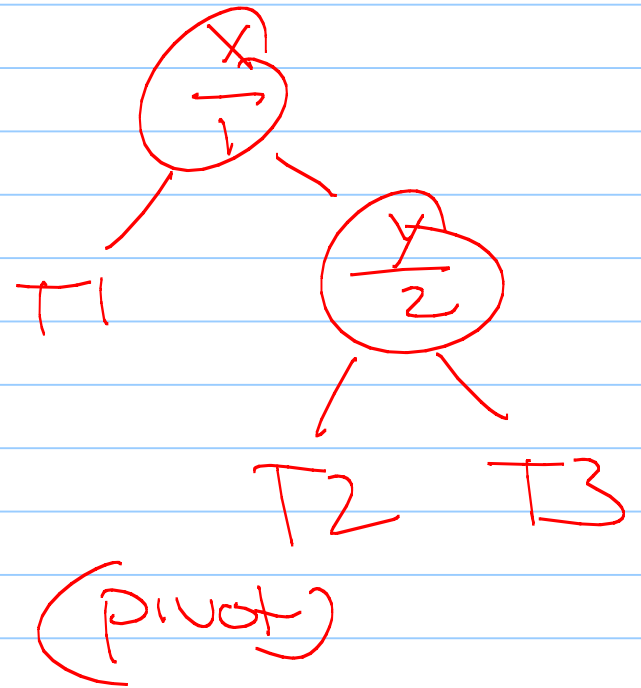


# Rotations

$x$  &  $y$  are in correct BST order, with  $x \leq y$ , but priorities are wrong

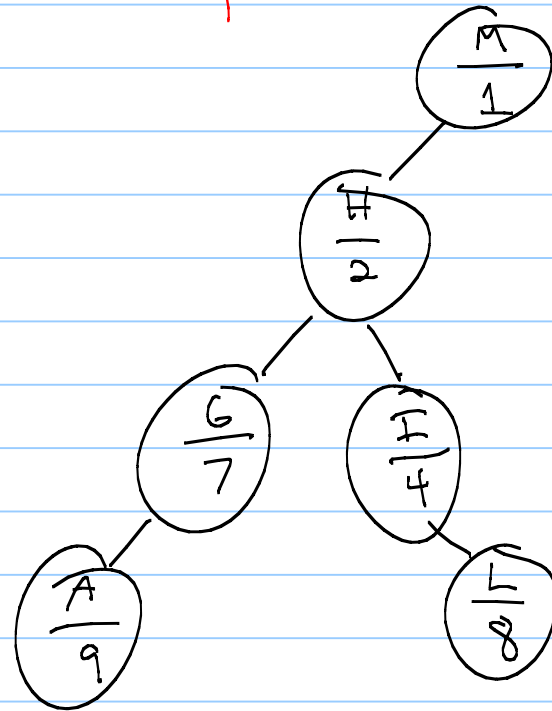
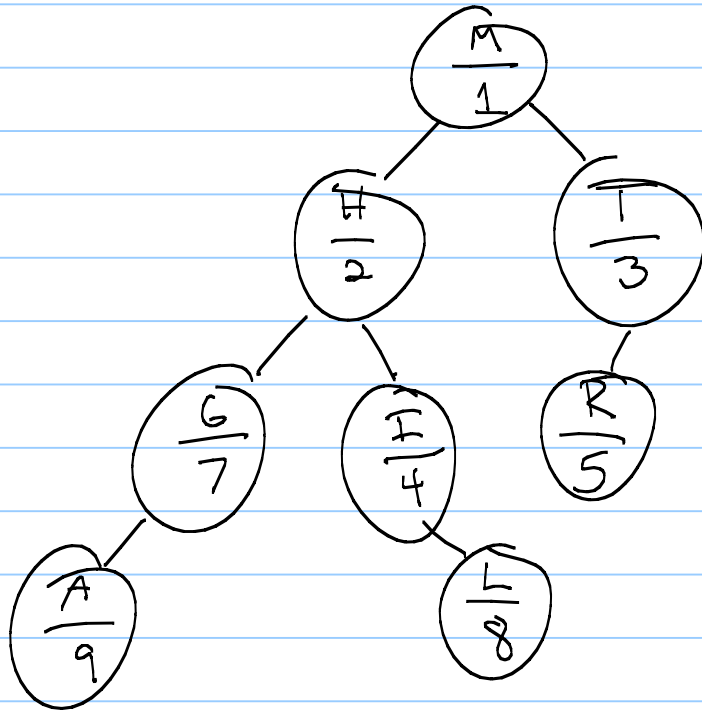


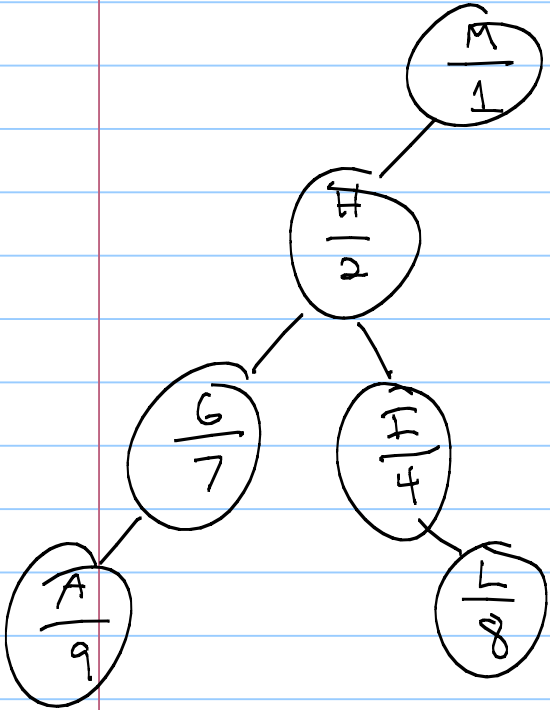
Fix:



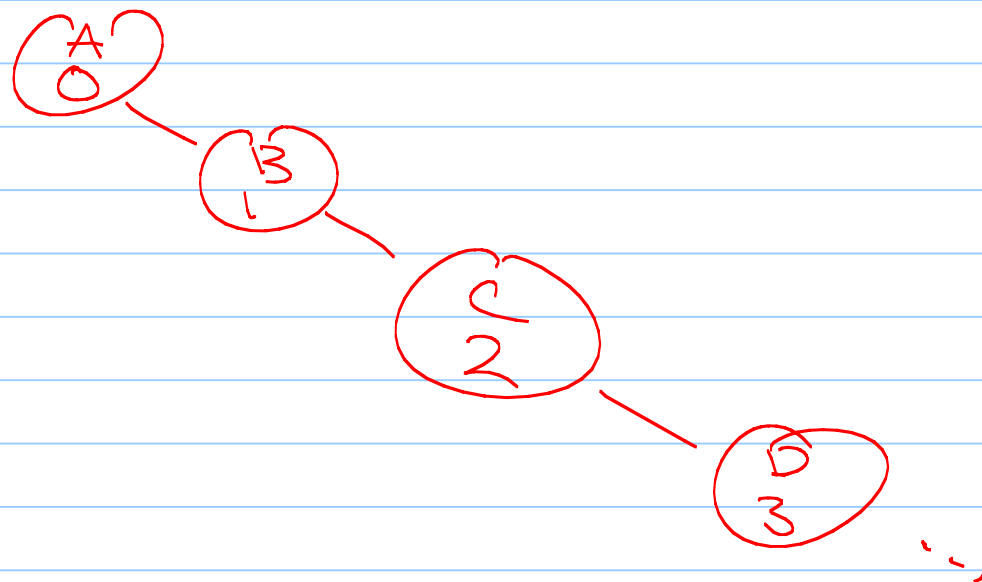
So: insert (S, 0)

(see prev. slides)





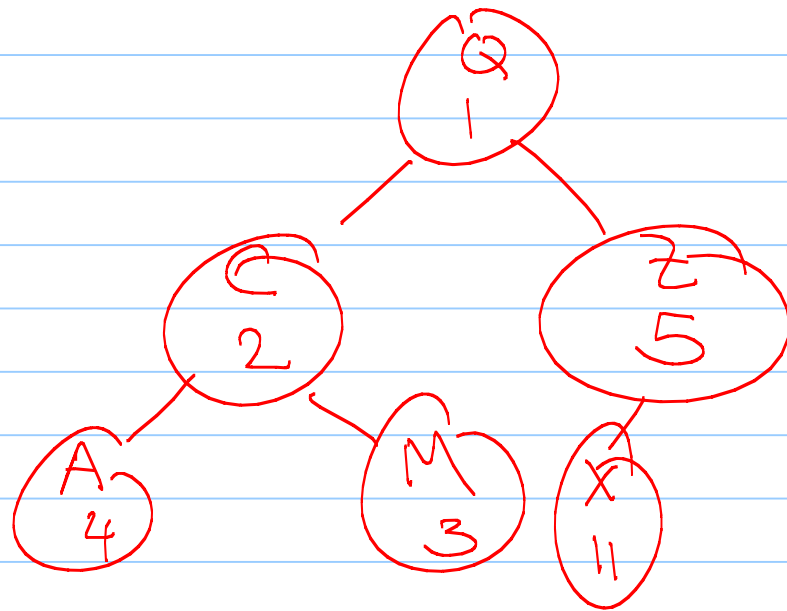
Downside: What can height be?  
Can we force them to be balanced?  $O(n)$



Treaps are Unique!

order of insertion doesn't matter.

Draw heap with ~~(A, 4)~~<sup>L</sup>, ~~(C, 2)~~<sup>L</sup>  
~~(X, 11)~~<sup>R</sup>, ~~(M, 3)~~<sup>L</sup>, ~~(Q, 1)~~<sup>R</sup>, ~~(Z, 5)~~<sup>R</sup>



## Randomized treaps :

Alternative to AVL trees.

Each element will get a random priority.

Expected height of the treap will be  $O(\log n)$ .



Code: How do we implement?

- Change node class

already coded pivot

↳ Inherit to new class  
+ needs find, insert + remove