

CS180 - Treaps

Note Title

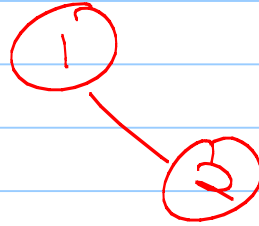
5/4/2011

Announcements

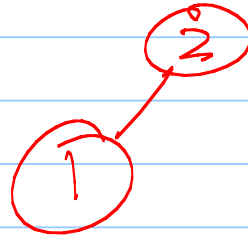
- Test Friday (review Thurs.)
- HW due tonight ←
- Next HW - up
- checkpoint in 1 week

Note on HW: #5

1, 2 a



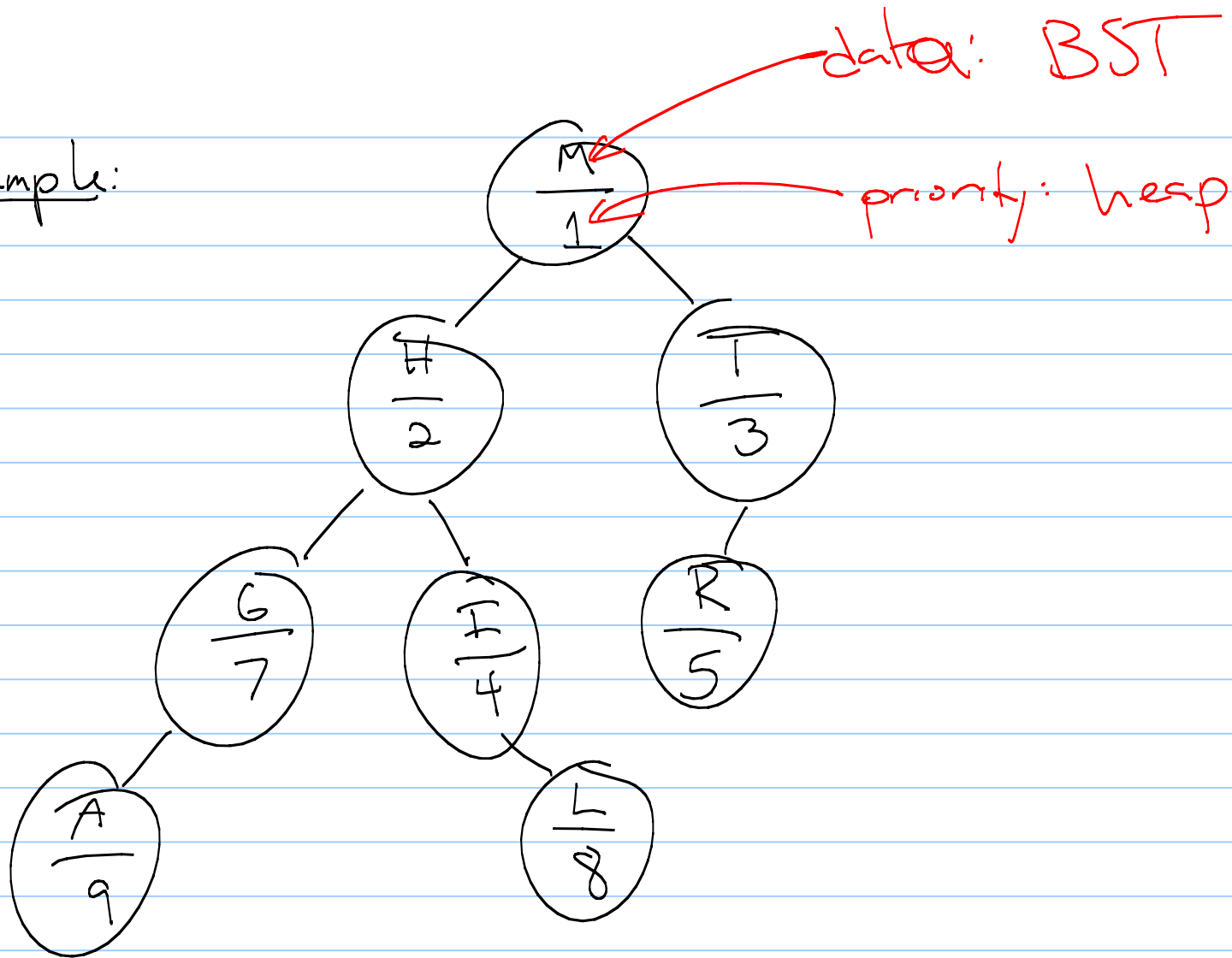
2, 1 a



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.
↑
min heap

Example:

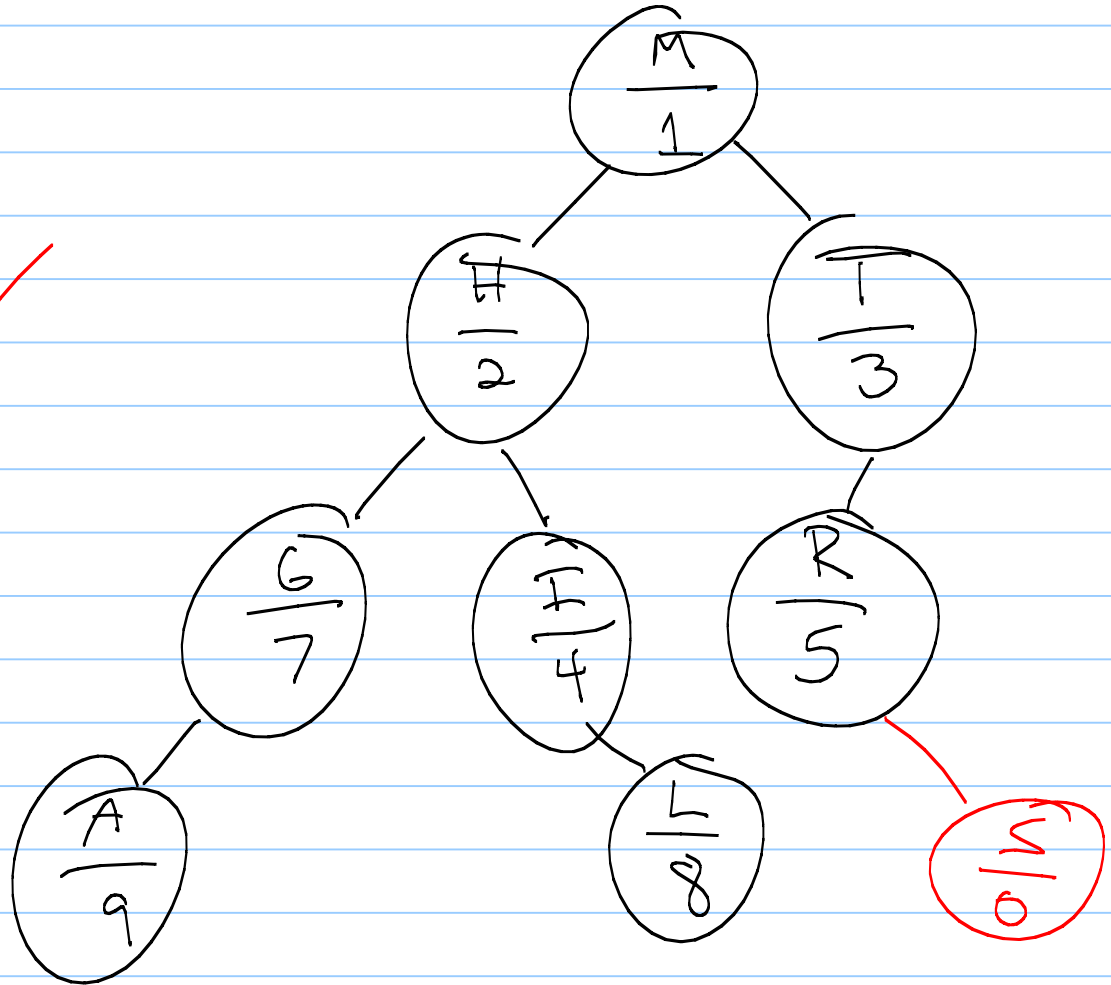


Insert

insert: (S, 0)

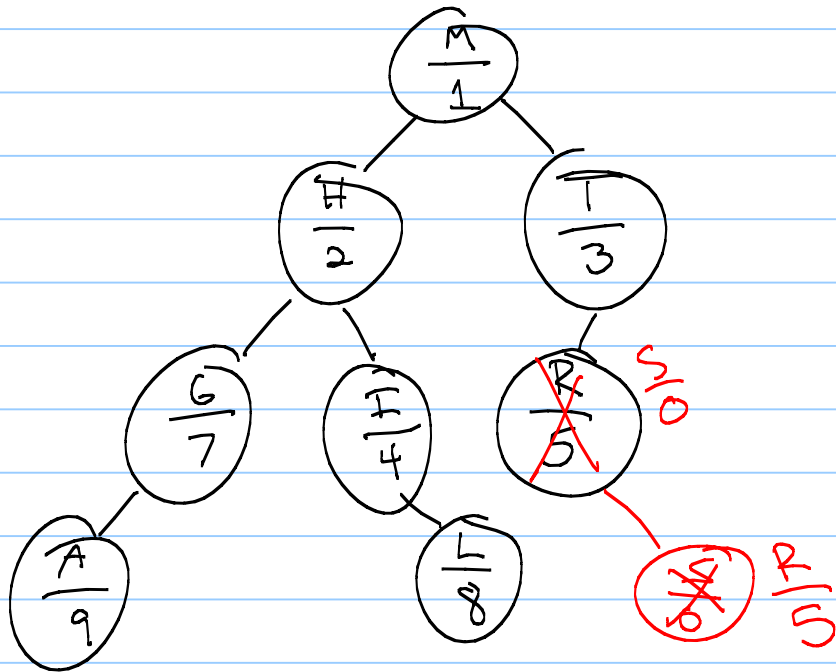
start BST: ✓

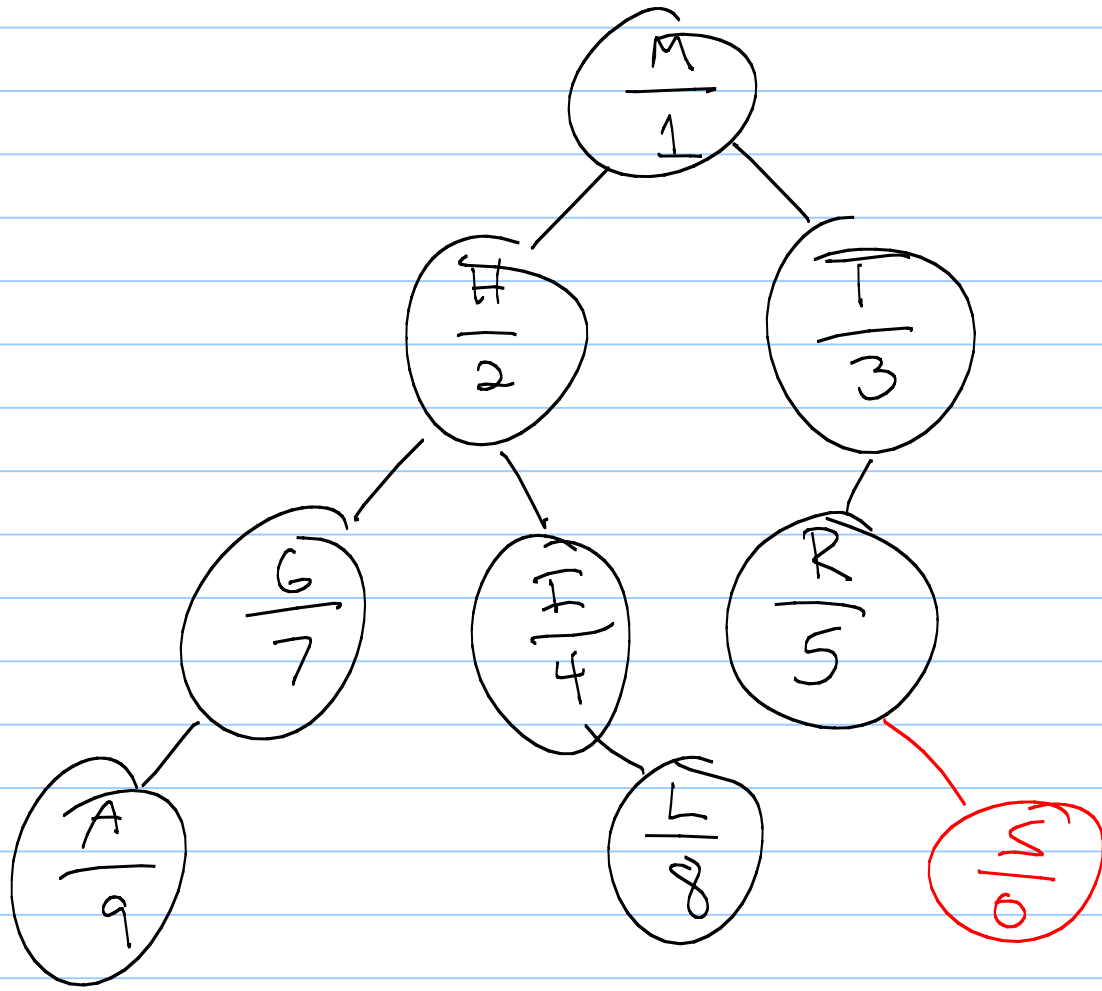
heap?



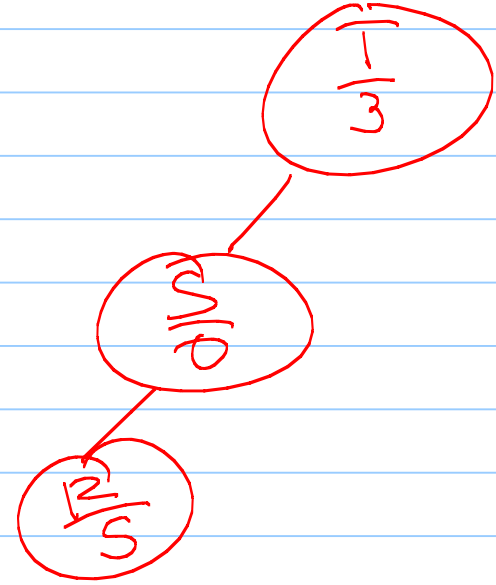
In heap, we "bubble up".
Will that work here?

No: bubbling
up may
break BST field.



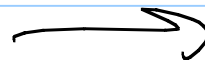
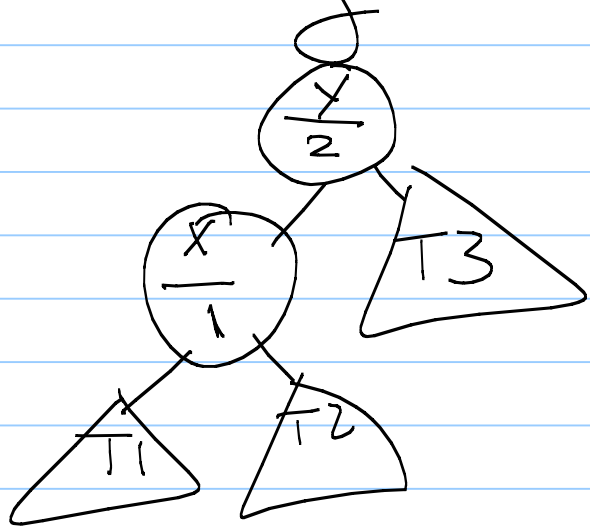


What to do?



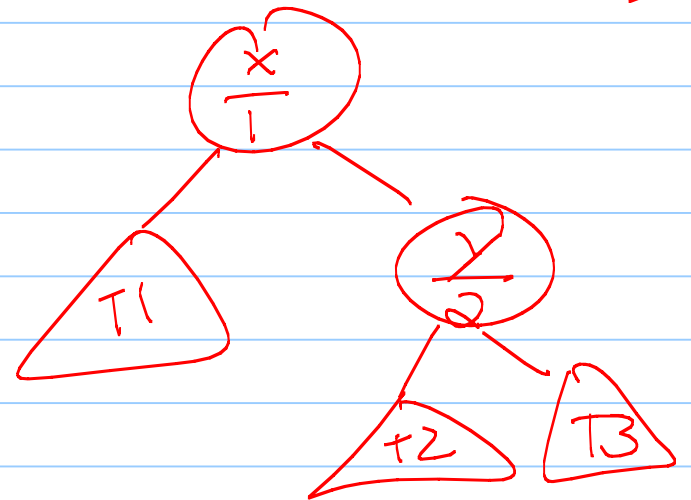
Rotations

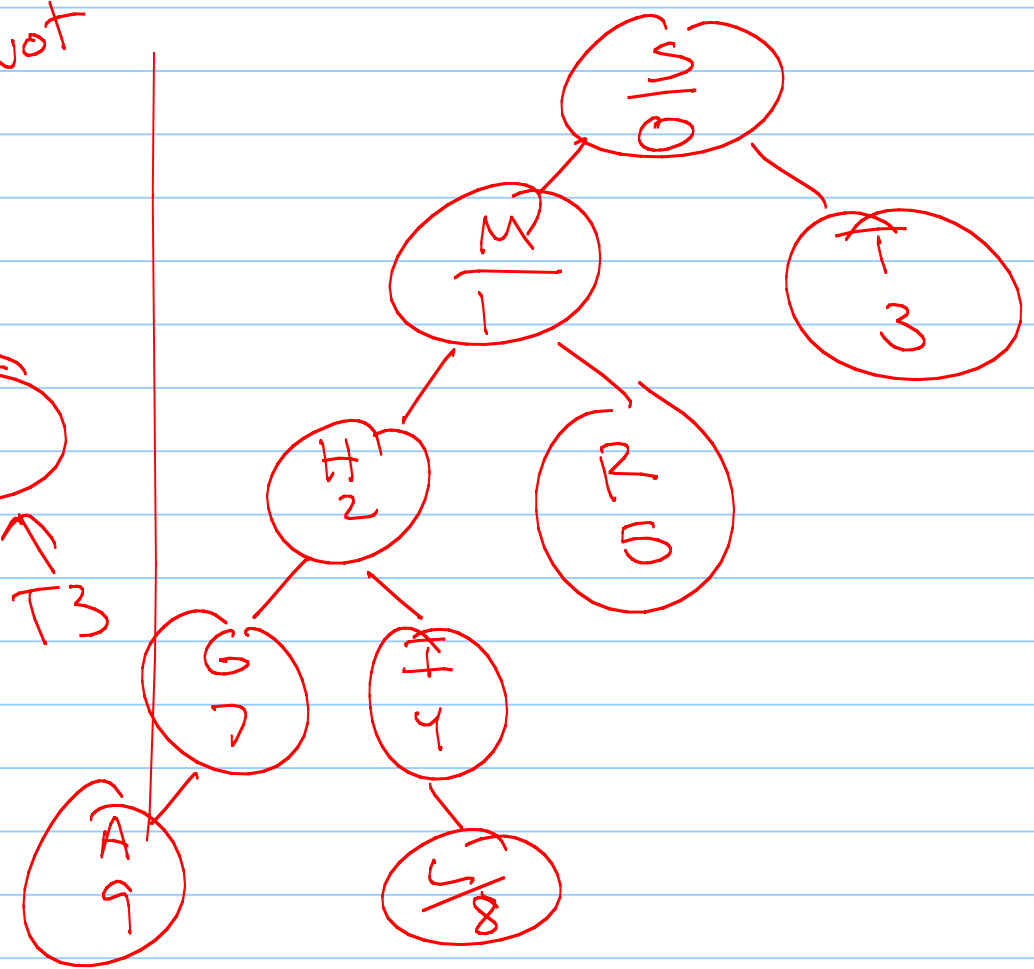
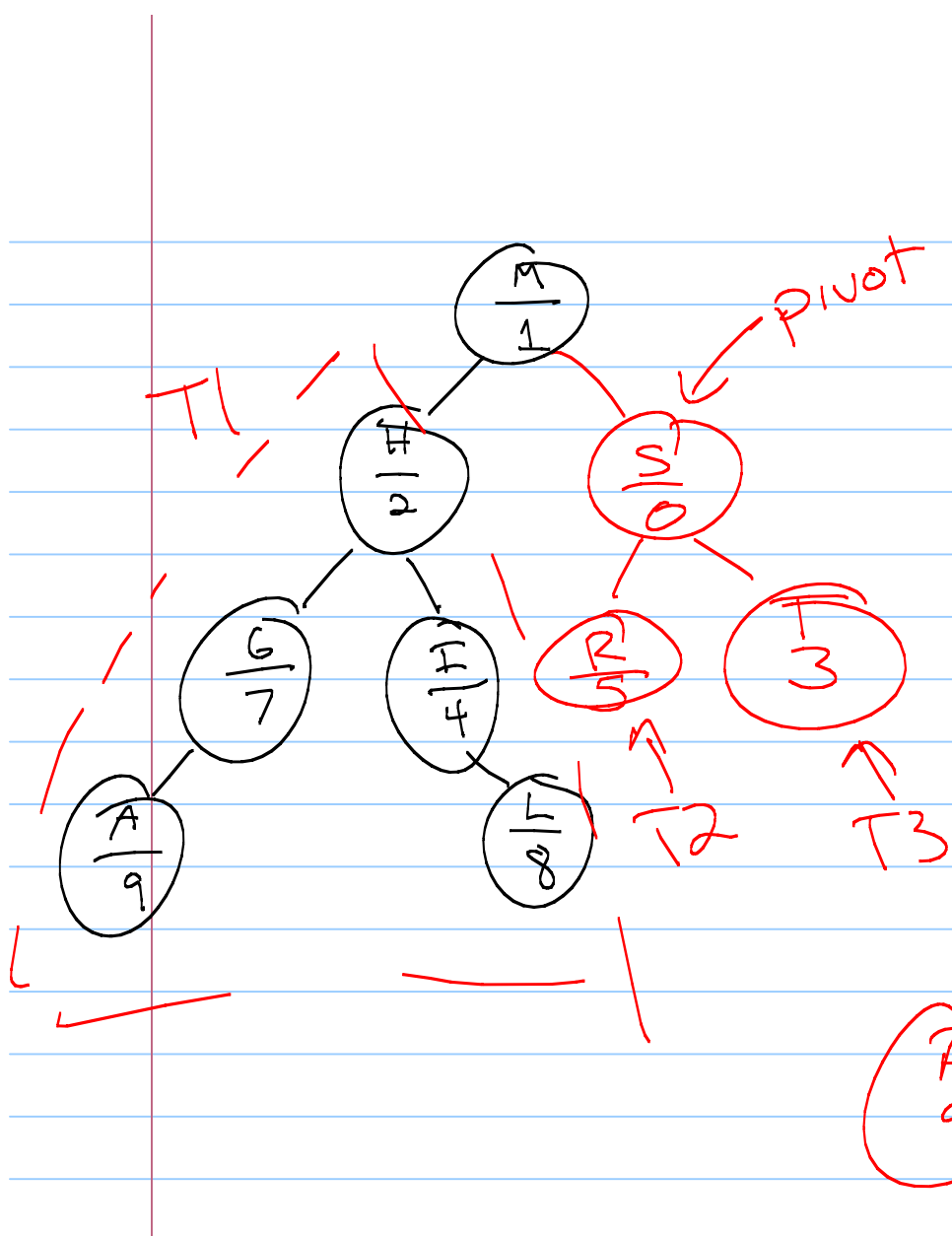
x & y are in correct BST order, with $x \leq y$, but priorities are wrong



Fix:

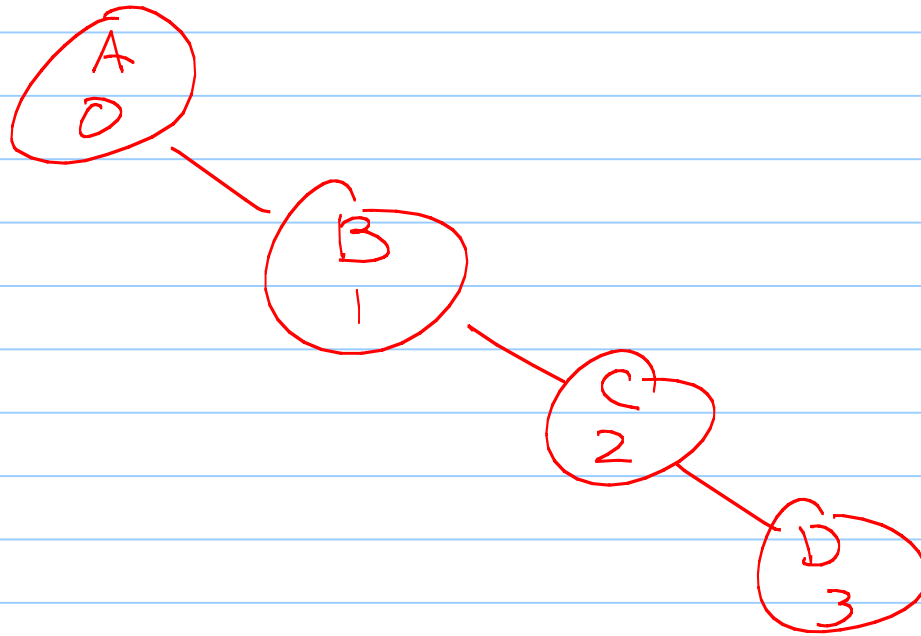
→ pivot (from Binary Tree class)



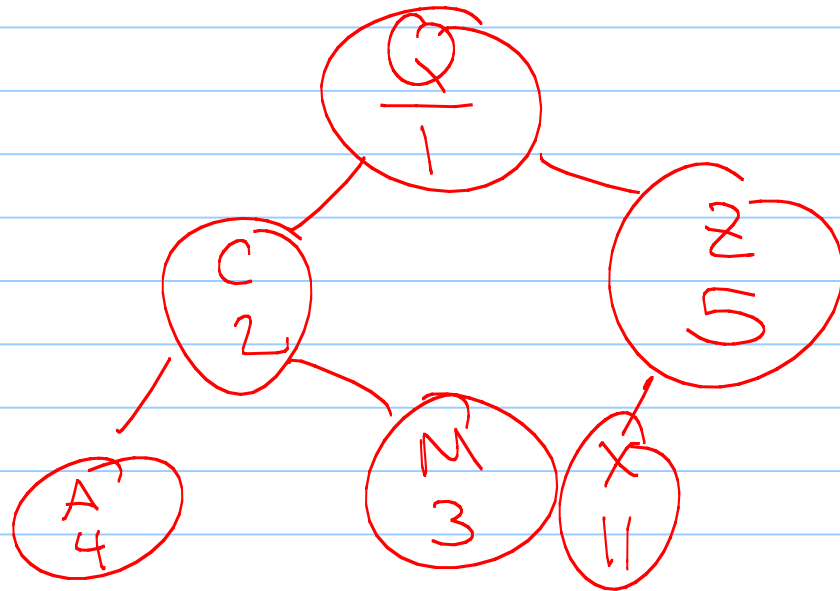


Downside: What can height be? $O(n)$

Can we force them to be balanced? **No**



Draw heap with ~~(A, 4)~~, ~~(C, 2)~~
~~(X, 11)~~, ~~(M, 3)~~, ~~(Q, 1)~~, (Z, 5)



Randomized treaps :

Alternative to AVL trees.

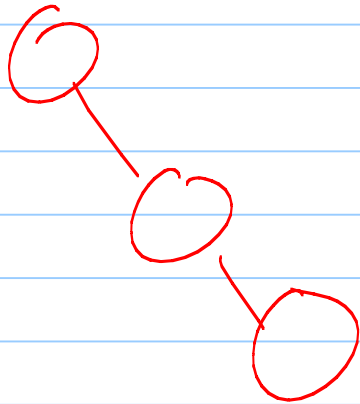
Each element will get a random priority.

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

Code: How do we implement?

- inherit from Binary Tree.h
(since not complete tree, like heap)

- code is actually easier than AVL
while (priority out of order)
pivot



Next time:
implement