CS180 - Treaps

Note Title 5/4/2011

Announcements

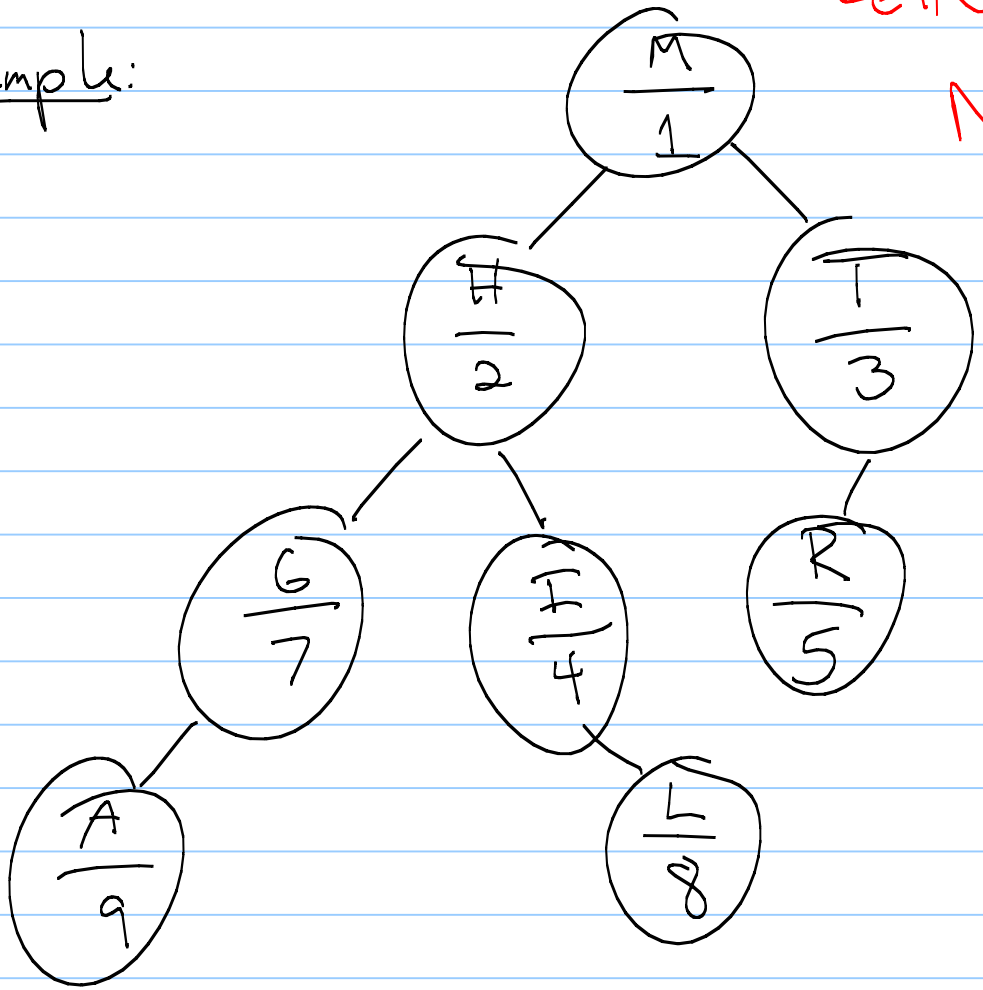
- Program due Friday
- Last HW will be up today
  due Monday
- Review Monday
- Final is Wednesday at noon
- Office hours next Tuesday

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.

tree heap = treap

Example:



Letters - data (BST)

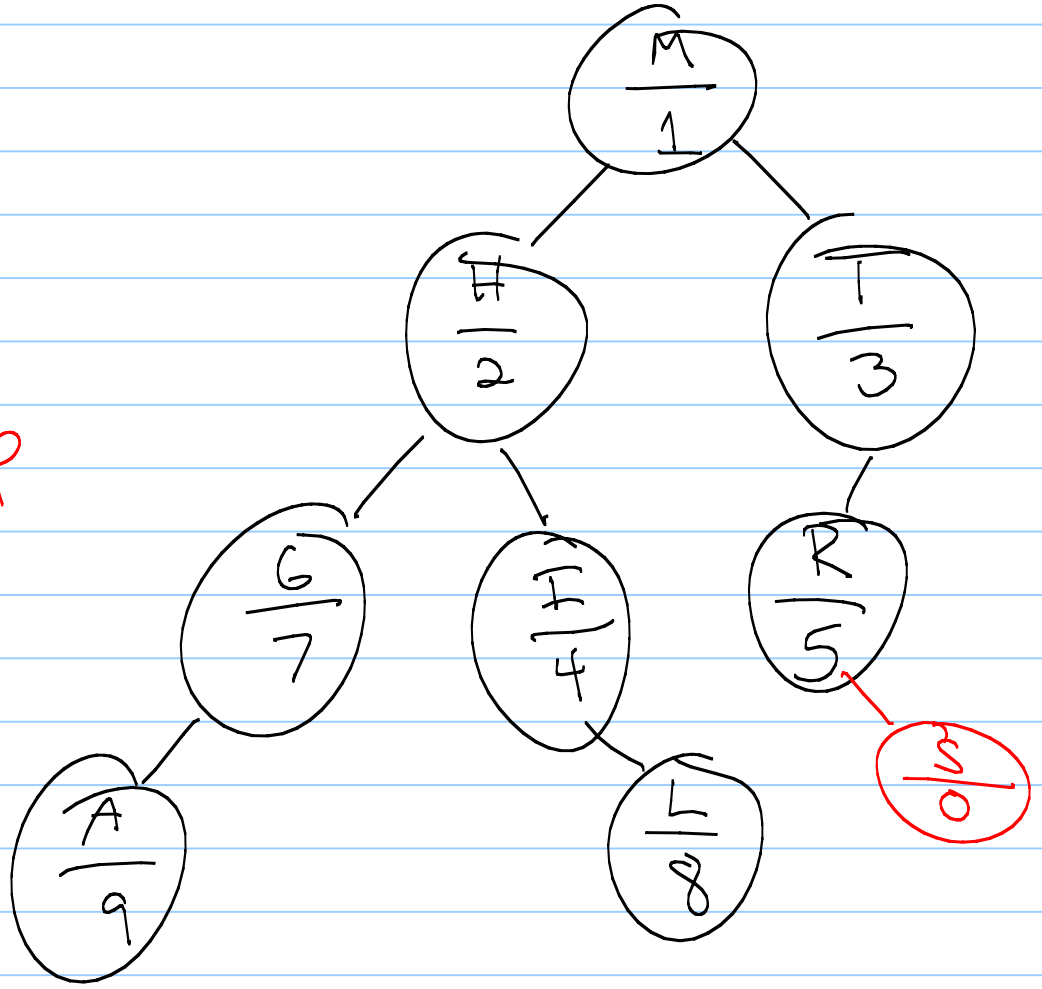
Numbers - priorities (heap)

min heap

Insert

insert: (S, 0)

Problem: violated heap property

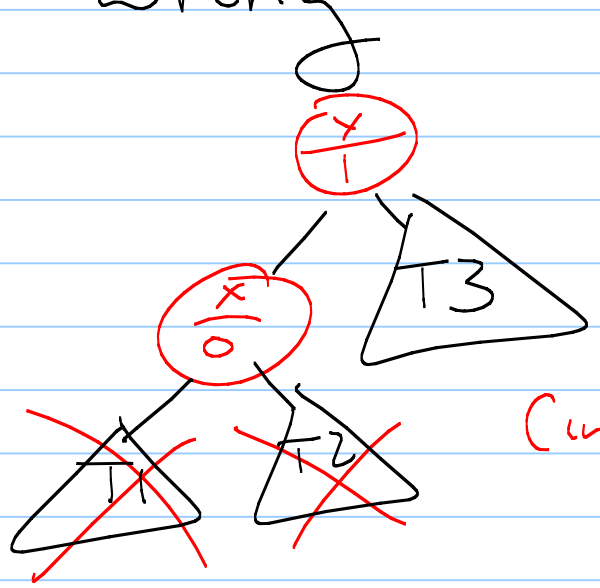


In heap, we "bubbled up".
Will that work here?

Here, need BST order.

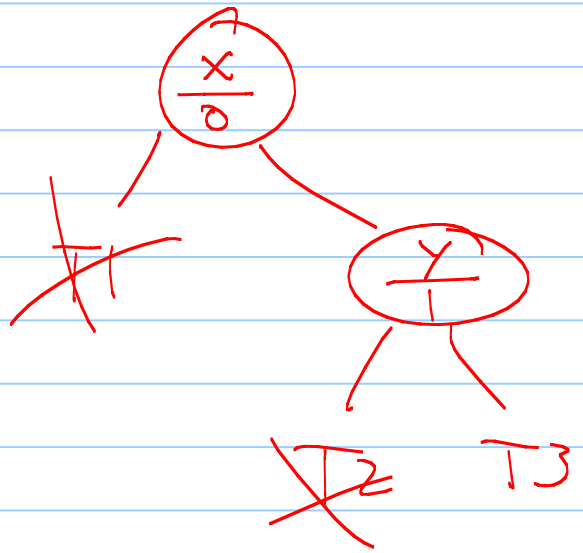
Rotations

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

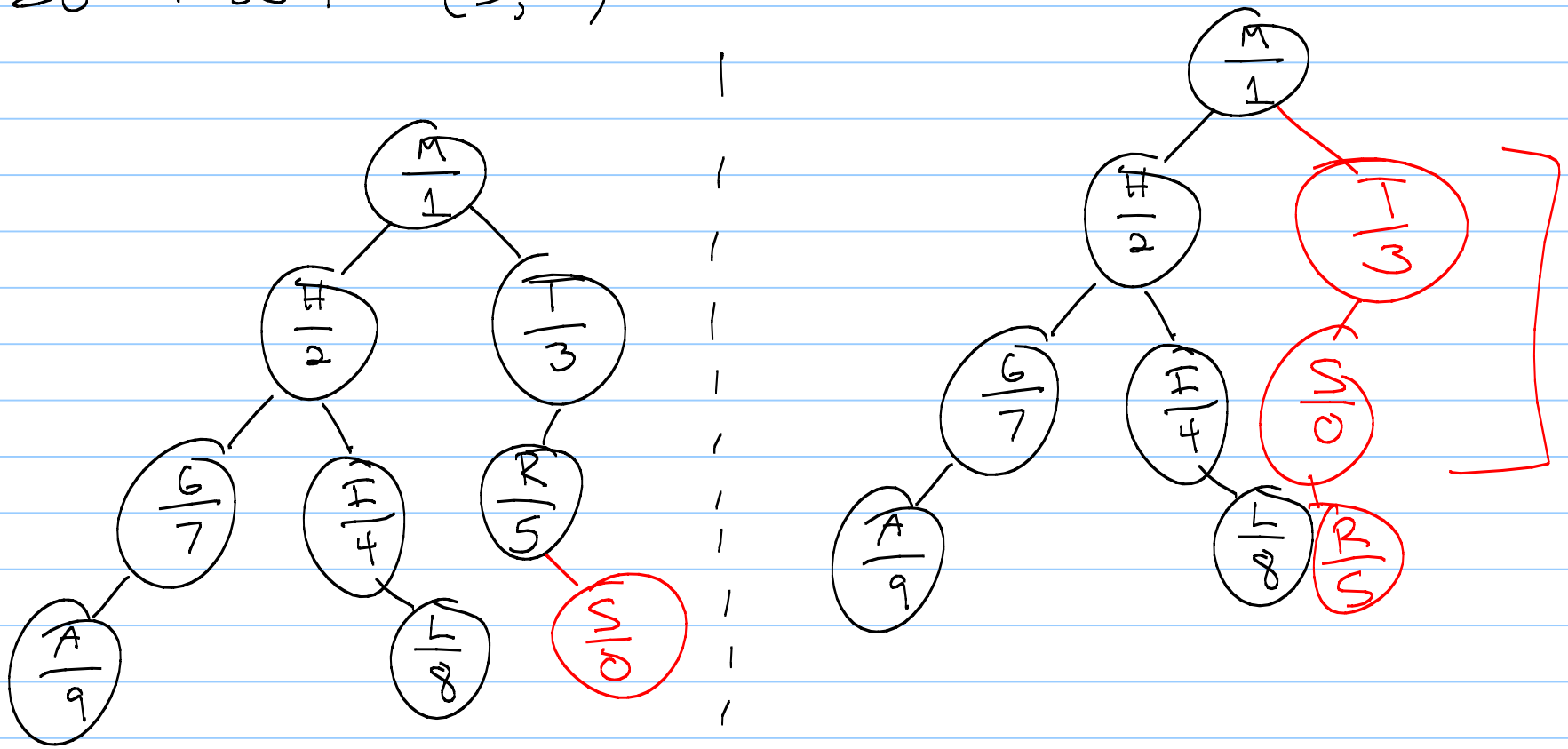


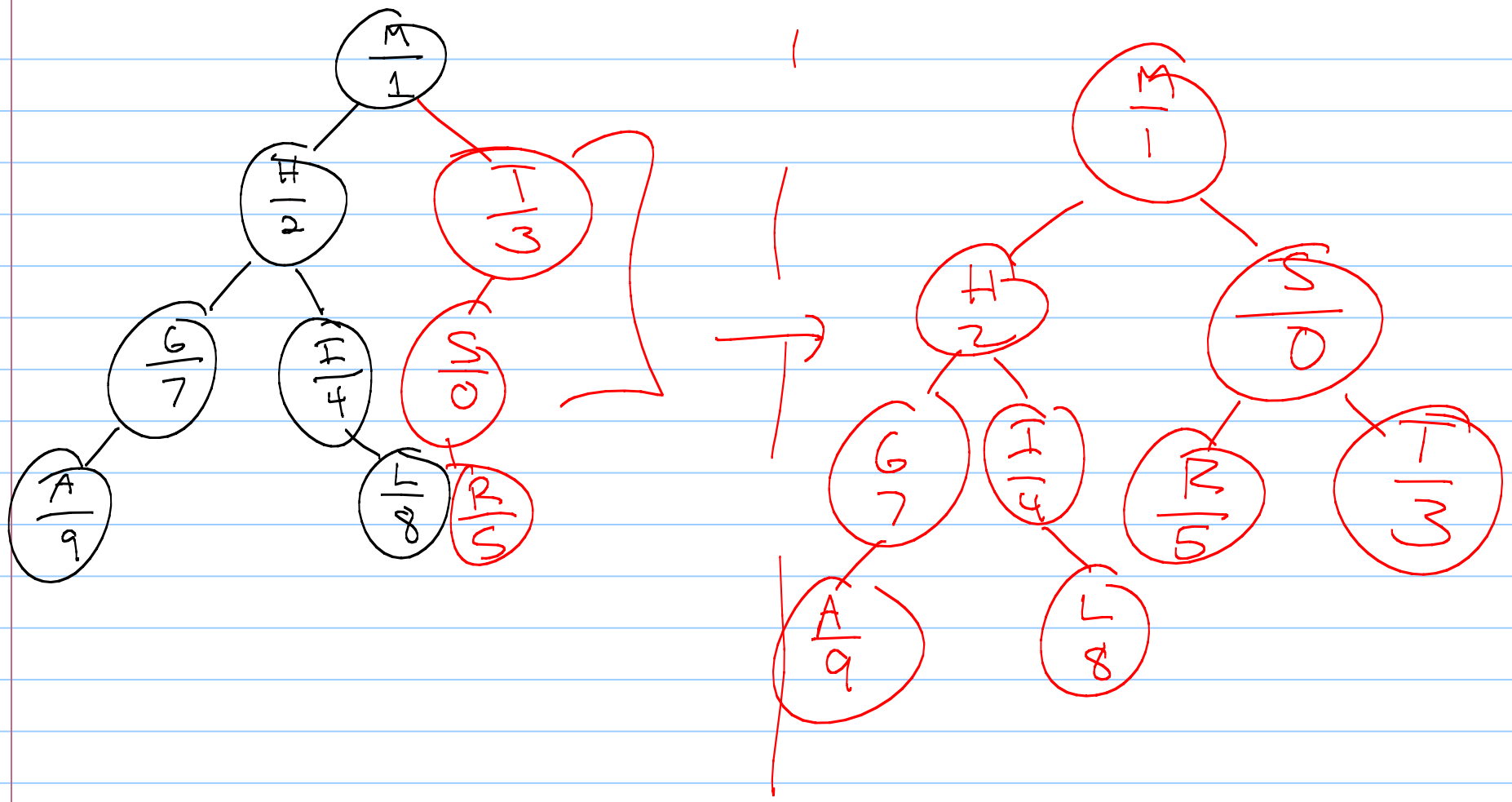
(insert x, 0)

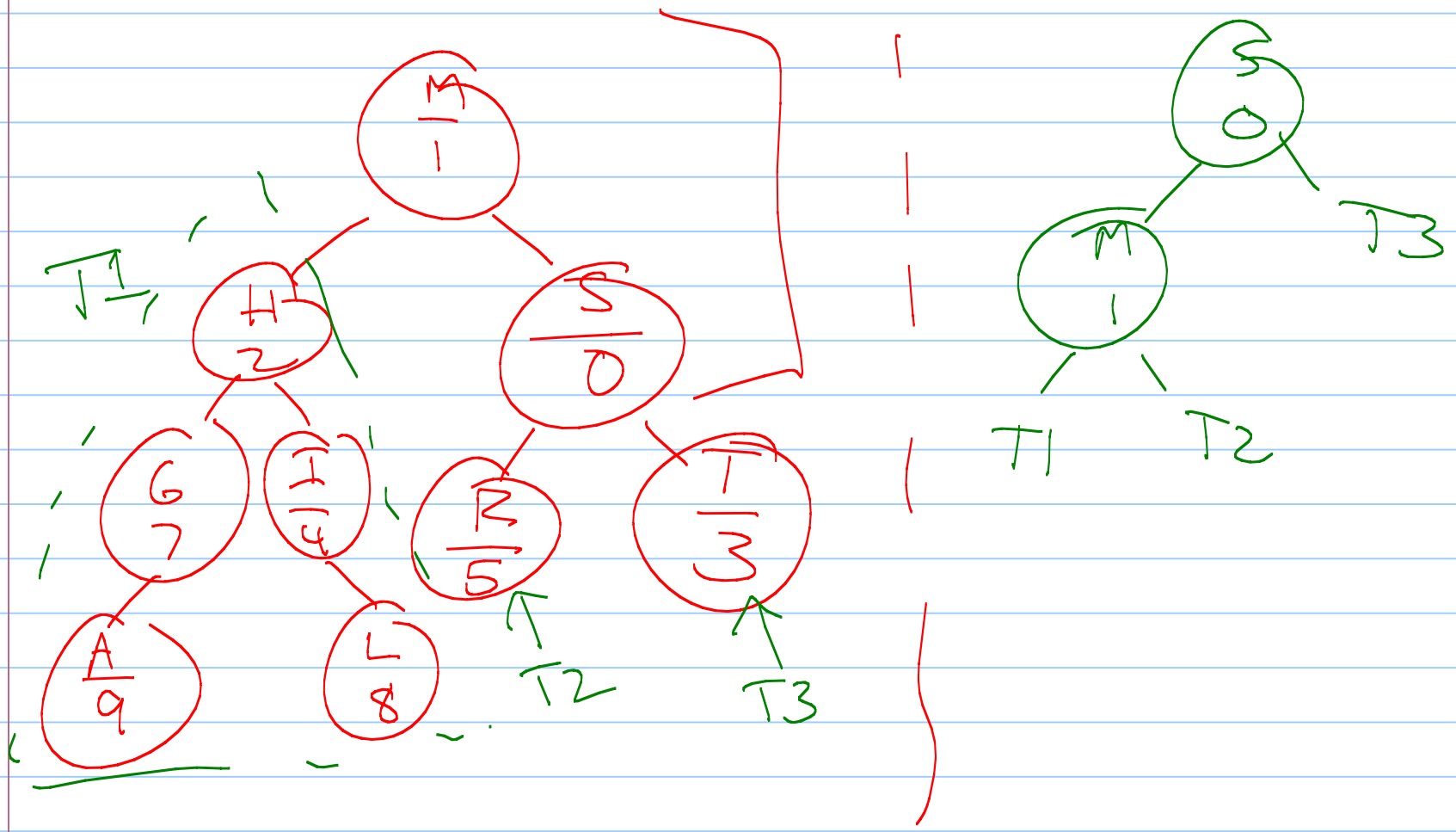
Fix:



S_0 : insert (S, 0)




M 1
H 2
T 3
G 7
I 4
S 0
A 9
L 8
R 5
M 1
H 2
S 0
G 7
I 4
R 5
T 3
A 9
L 8


M
1
H
2
S
0
G
7
I
4
R
5
T
3
A
9
L
8
T1
T2
T3
S
0
M
1
T3
T1
T2

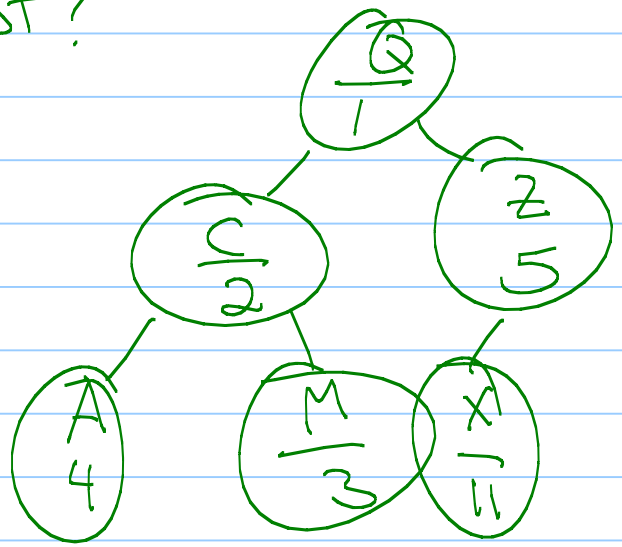
Downside: What can height be? O(n)

Can we force them to be balanced? No

Any set of nodes gives unique treap (no matter how inserted)

Draw treap with $(A, 4)^L$, $(C, 2)^L$, $(X, 11)^R$, $(M, 3)^L$, ~~$(Q, 1)$~~, $(Z, 5)^R$

Who is the root?



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 BST

_aux becomes priority

find is same

insert - use rand() to set _aux
& use _pivot