CS 2100 - Treaps

Note Title 5/4/2011

Announcements

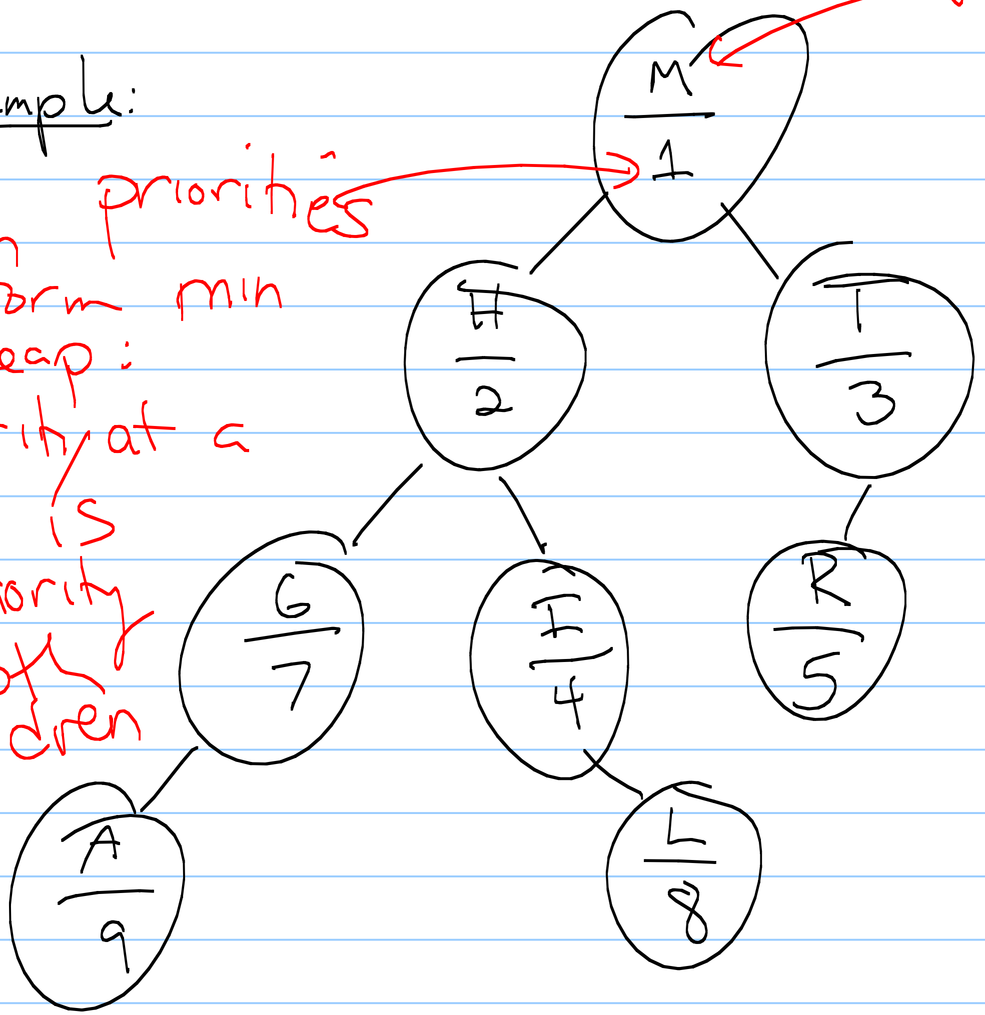
- HW 4 was emailed back to one partner (if you had a partner) yesterday
  (Email me w/ any issues)

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 heaps

Example:



priorities

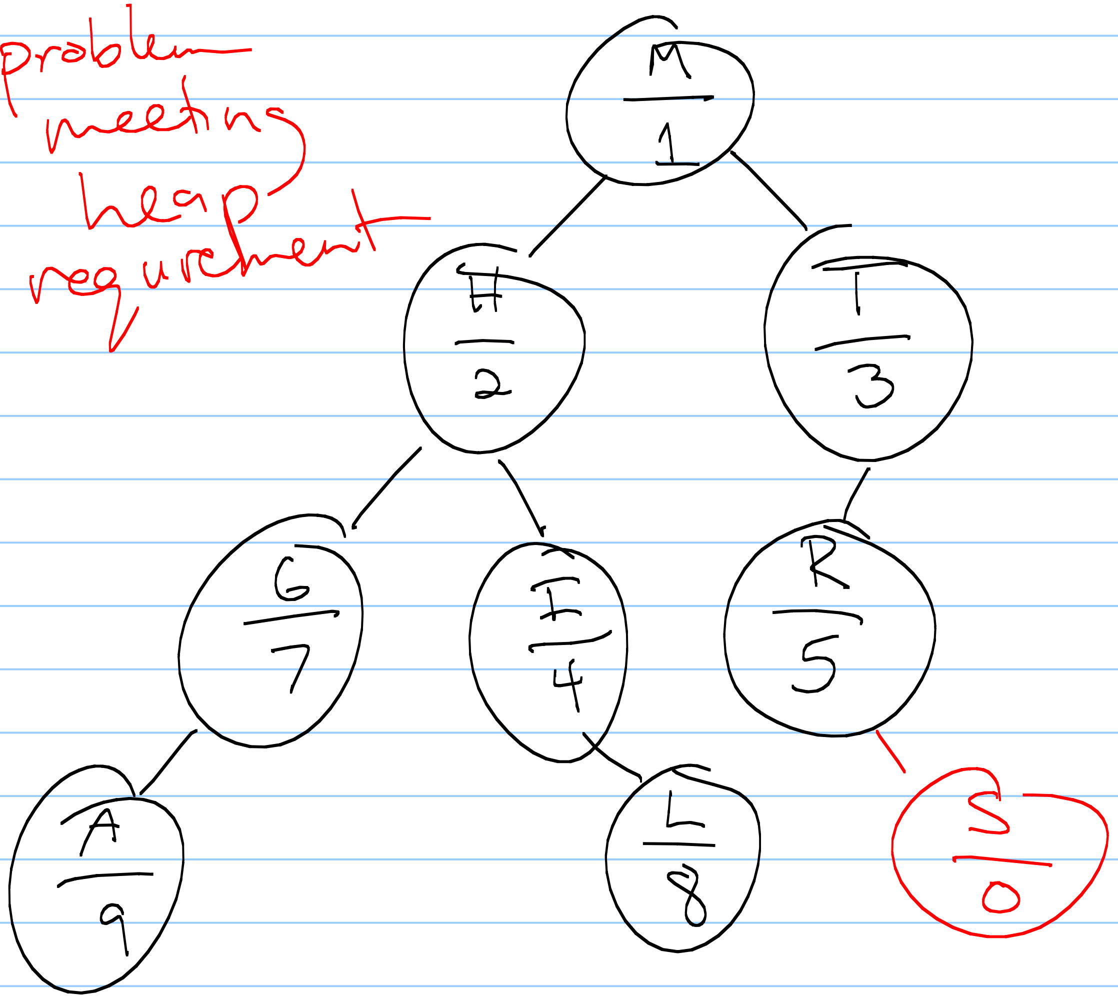
values (alphabetical order as my comparison for BST)

form min heap: priority at a node is ≤ priority at both children

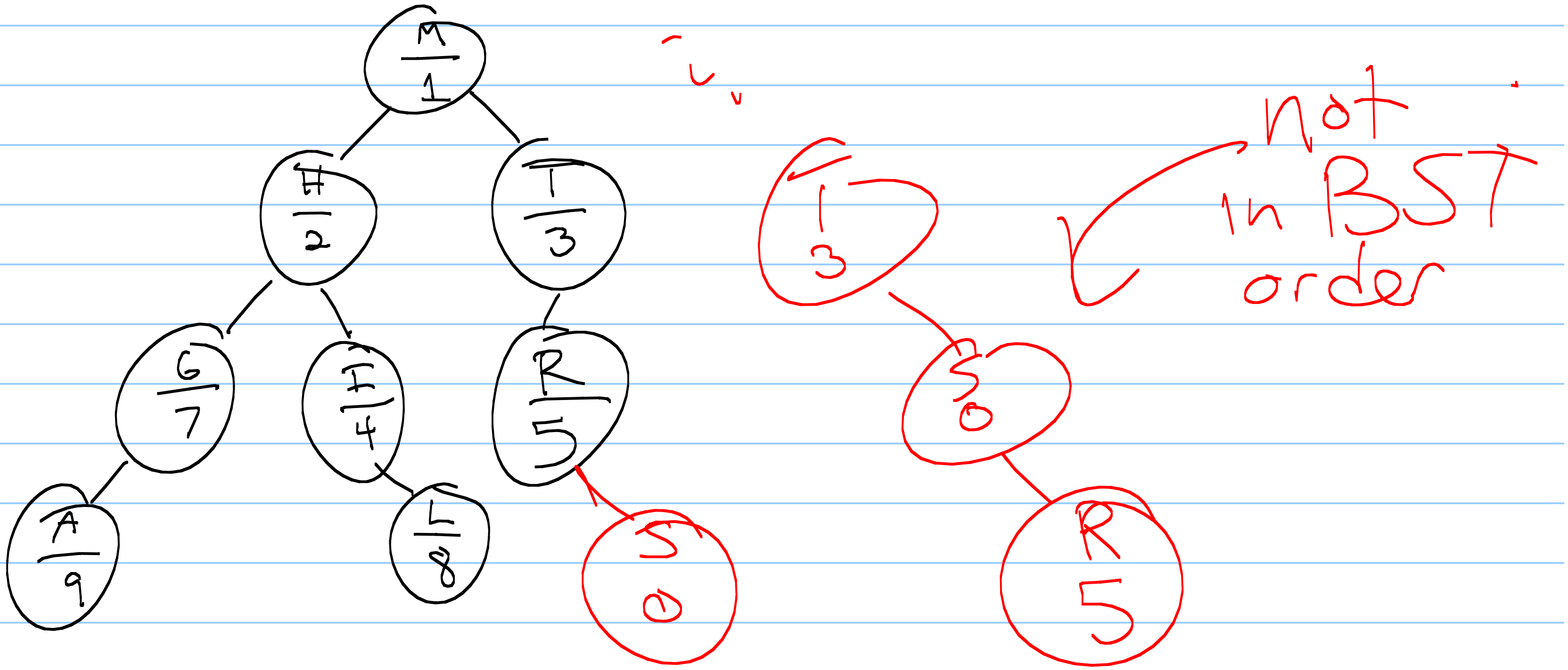
Insert

insert: (S, 0)

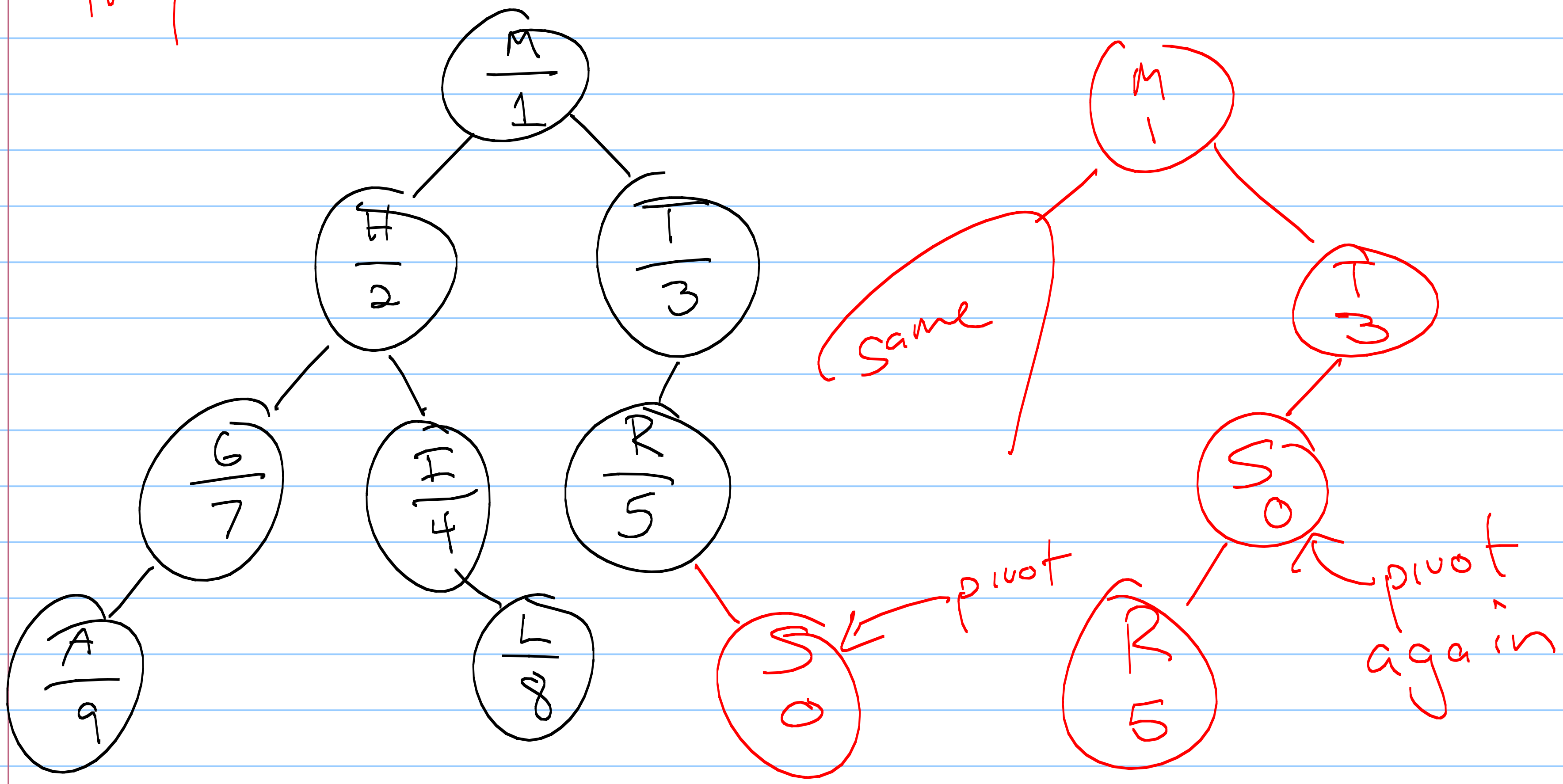
problem meeting heap requrement

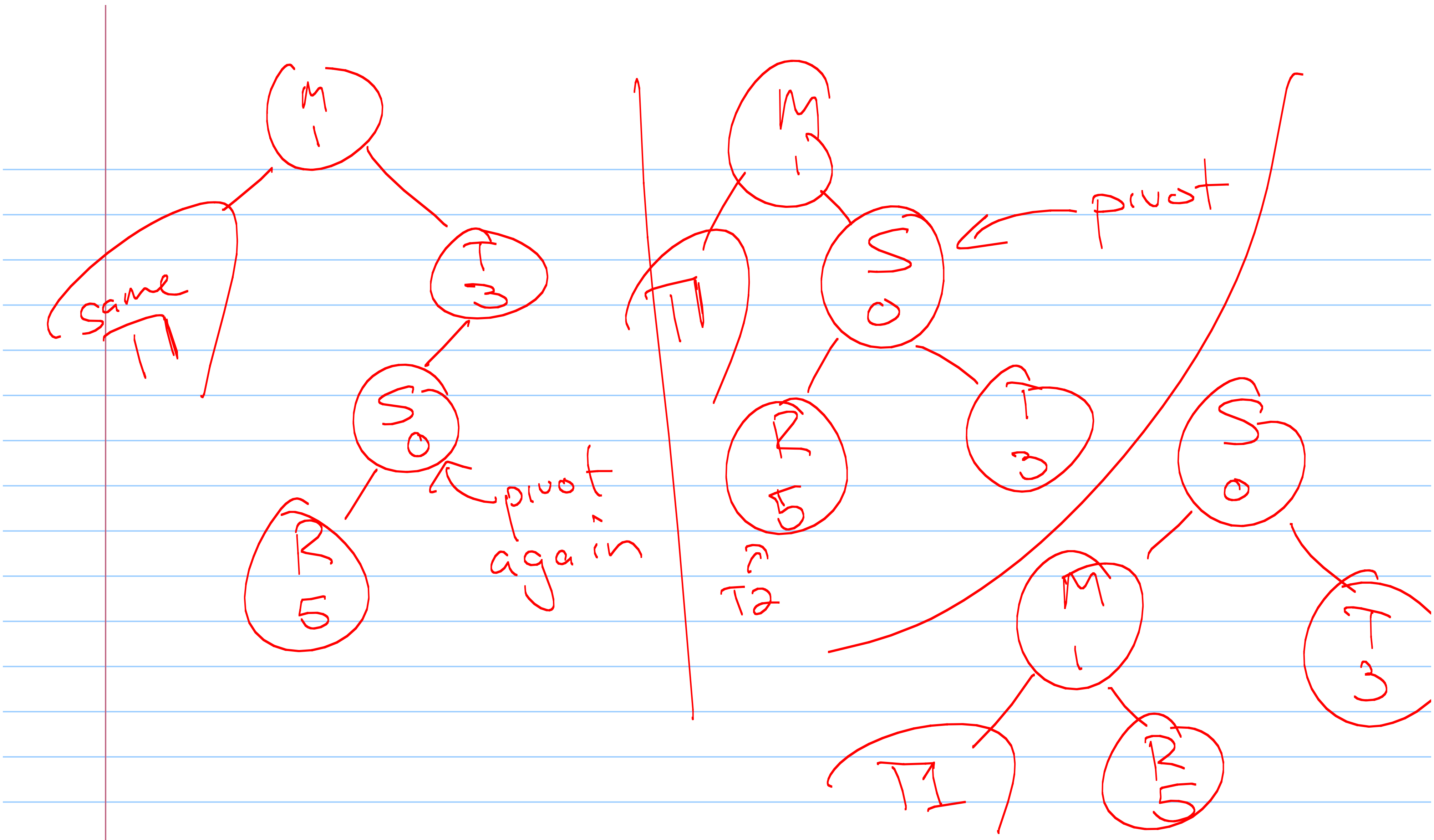


In heap, we "bubbled up".
Will that work here?



Try AVL operation: pivot

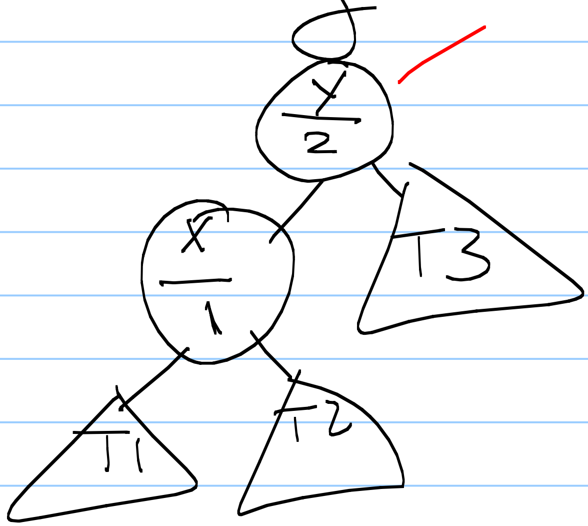


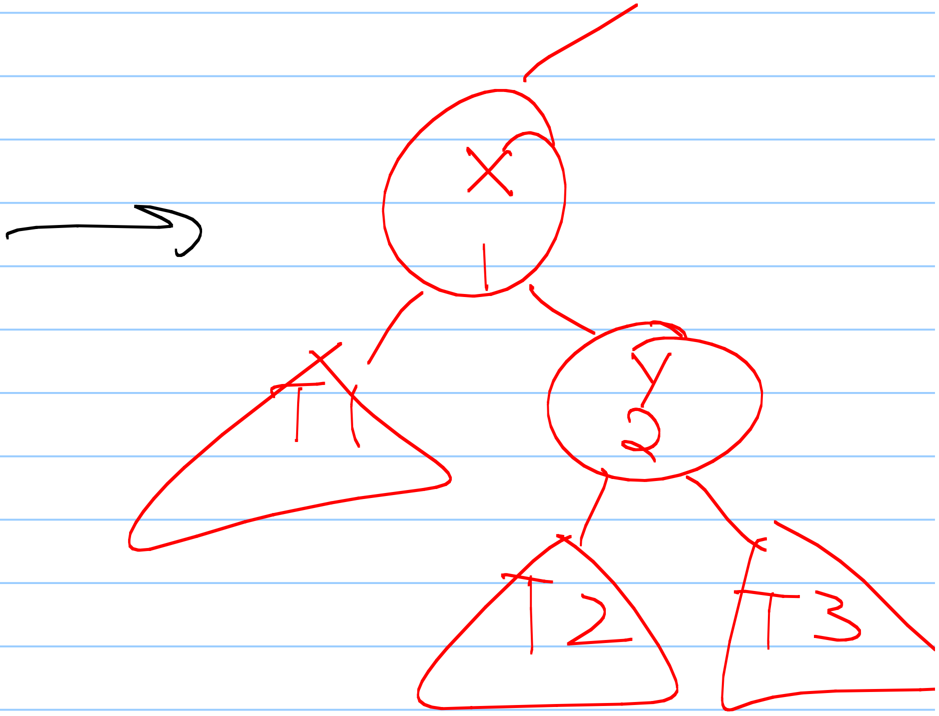

M 1
T 3
S 0
R 5
same T1
pivot again
M 1
T1
S 0
pivot
R 5
T 3
T2?
S 0
M 1
T 3
T1
R 5

Rotations

x & y are in correct BST order, with x ≤ y, but priorities are wrong

Ex:





So : insert procedure :

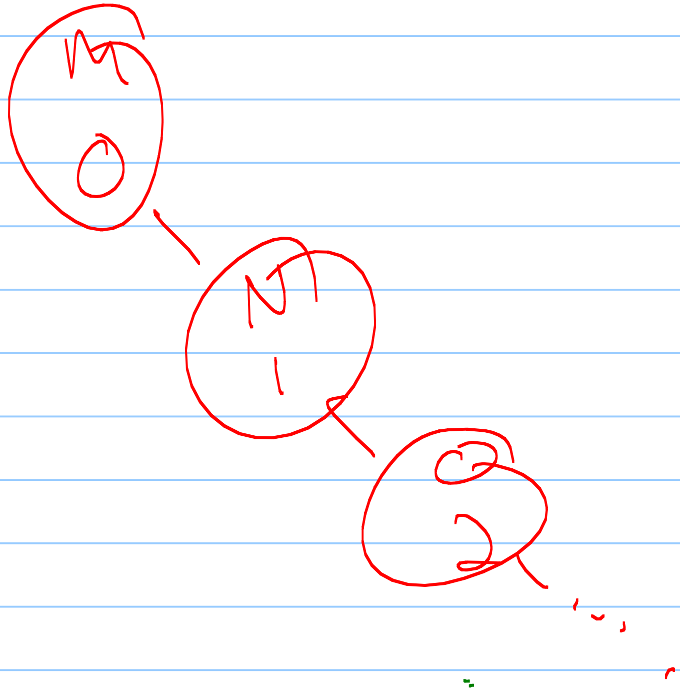
Insert as in BST using value

while (my priority < parent's priority)

pivot

Downside: What can height be?

↳ Can we force them to be (un)balanced?

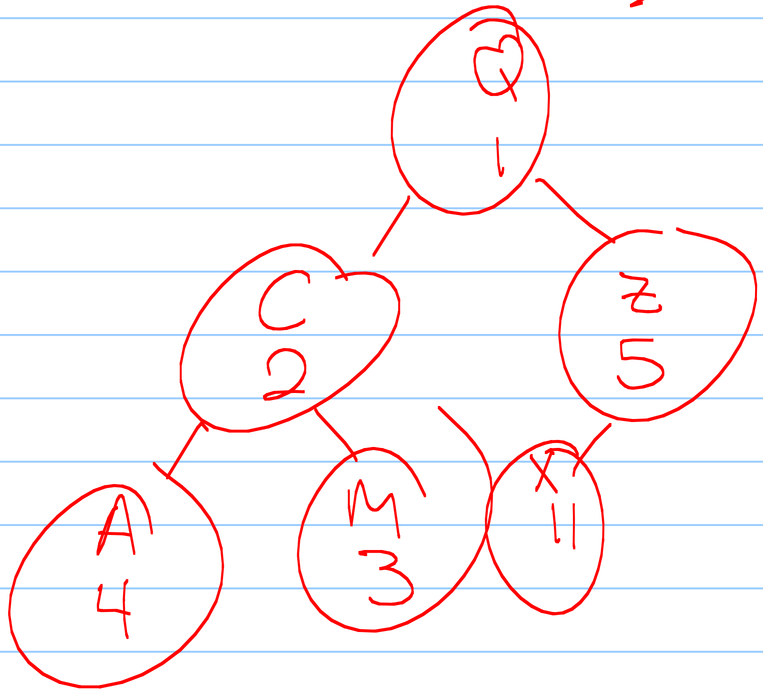


In fact: Treaps are unique! Order of insertion does not matter:

- if you try to change height of a node, will violate heap property
- if you try to move position, will violate BST property

(Like having 2 traversals)

Ex: Draw treap 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 $O(\log n)$.

Insertion: Q, A, L, M, Z, R, . . .

random priorities → 13 82 50 46 . . .

Code: How do we implement?

Inherit from BinaryTree.h

(or BinarySearchTree.h)

aux to be float (?)

- find – same as BST
- insert – 5 slides back
- remove: