CSCI 2100

Treaps

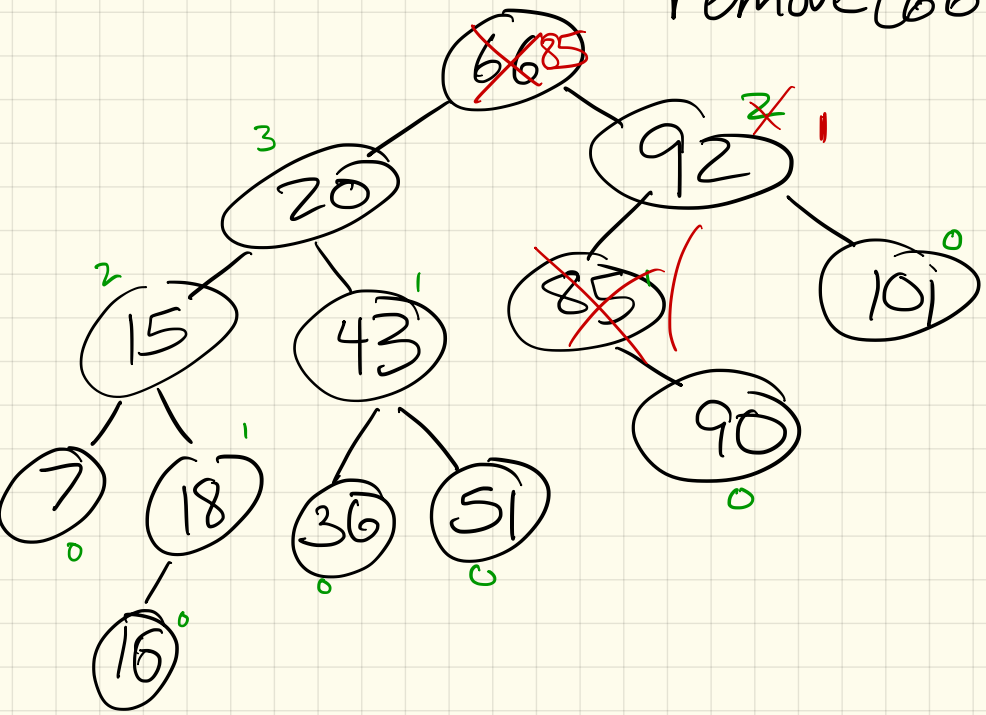
Recap

- Reading on ZyBooks for Wed. & Friday.
- Lab (posted) - due Sunday (prelab on Thurs as usual)
- HW due Monday, review in class
  ↖ no extensions
- Exam 2 next Wednesday (with lab the day after - I'll be back by then)
- Be nice to my subs please!! ☺

Remove in an AVL:

(last quick coverage)

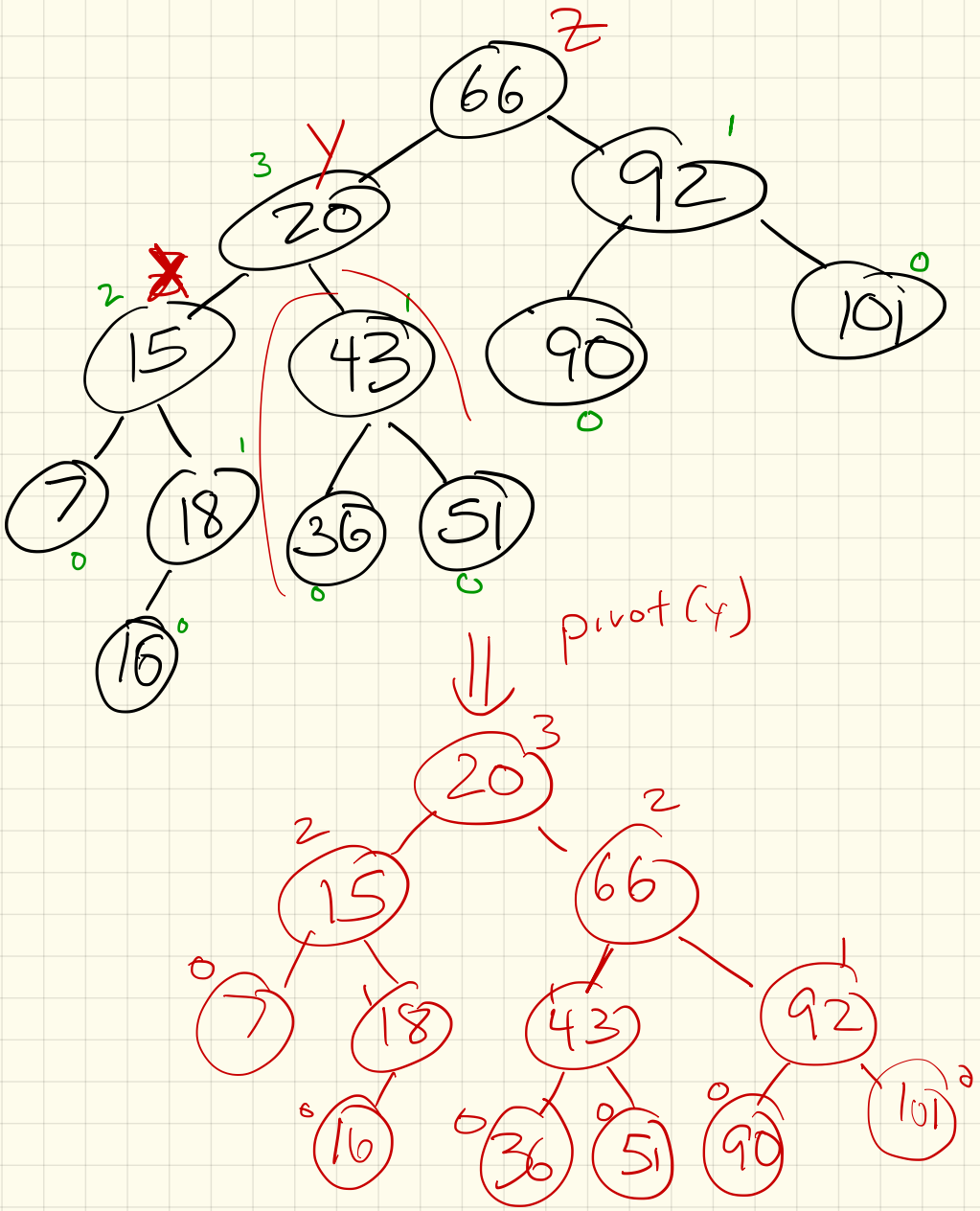
remove(66)



First: BST remove, re-do heights

Then: rebalance!

(if unhappy) ⤳


z
y
x
66
20
92
15
43
90
101
7
18
36
51
16
pivot(y)
20
15
66
7
18
43
92
16
36
51
90
101

Treaps: a new binary tree structure

Goal: Each node will contain a *value* (like a BST) and a *priority* (like a heap).

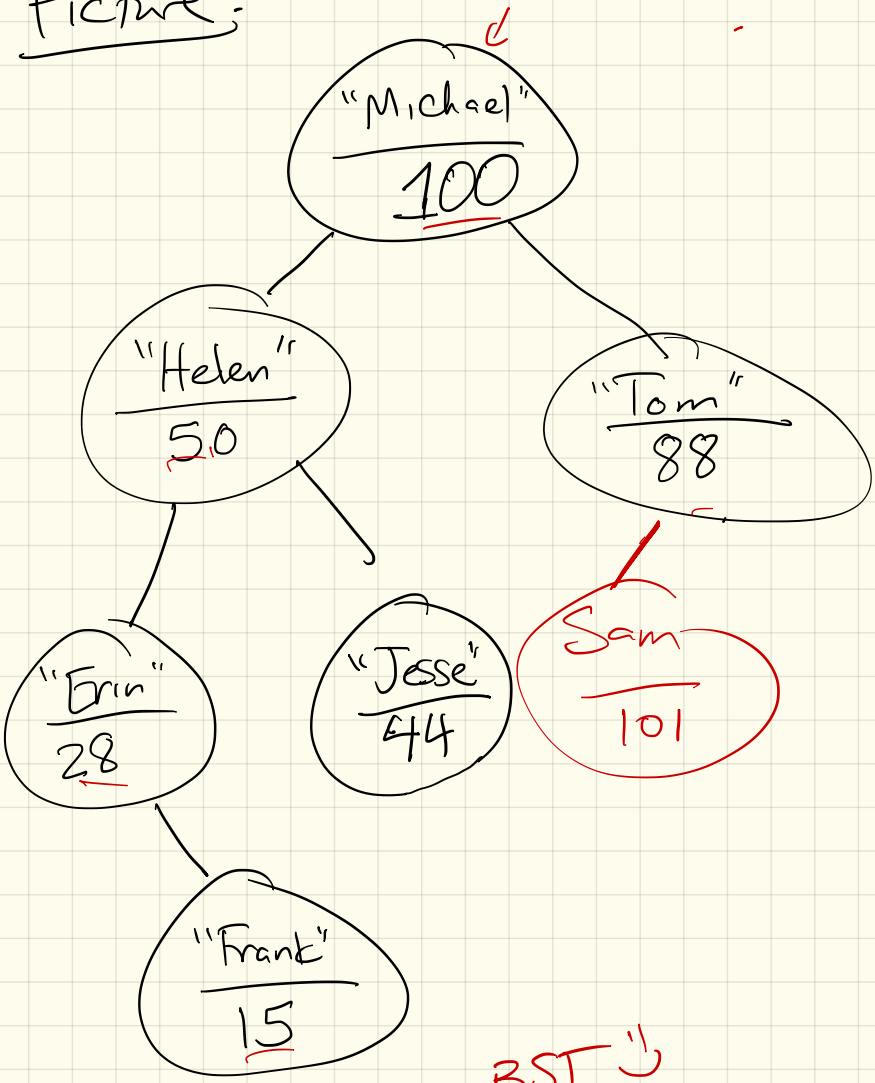
- BST over values
- heap over priorities

Ex: Suppose values are names and priorities are integers.

Both can be "sorted":

- values/names have alphabetical order
- integers (obviously)

Picture:

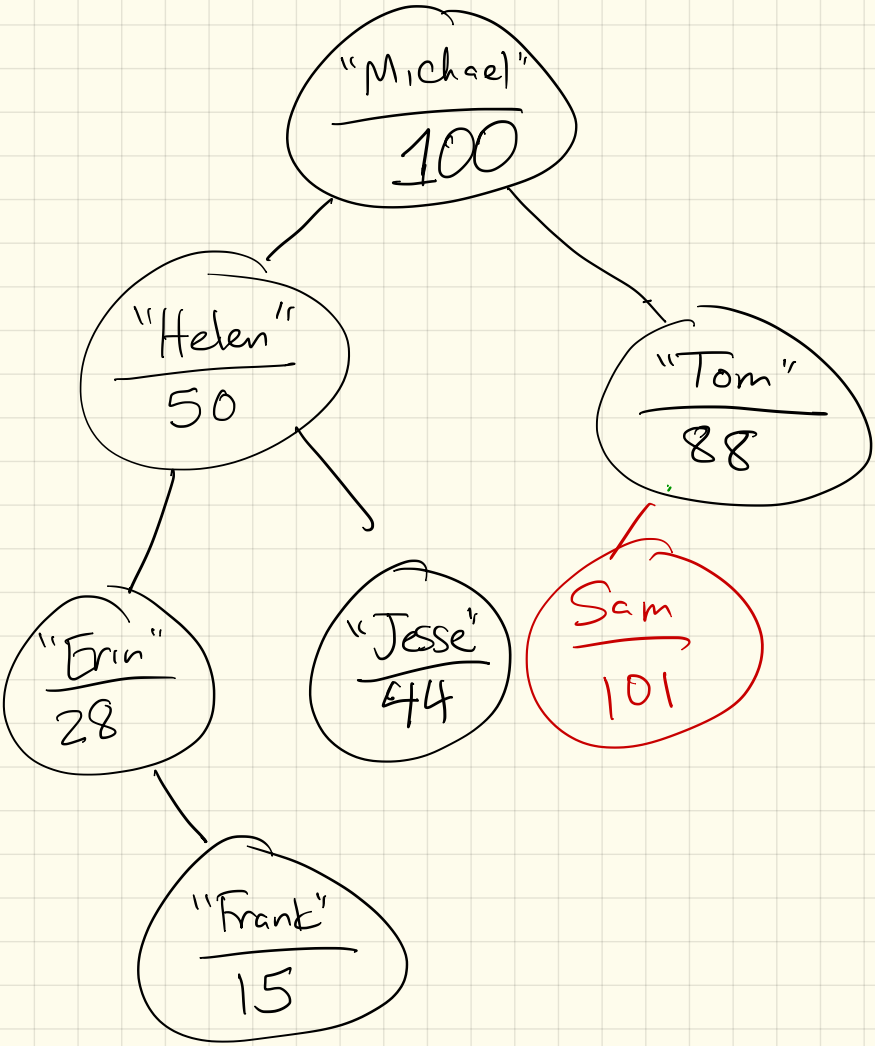


BST :)

Heap :(

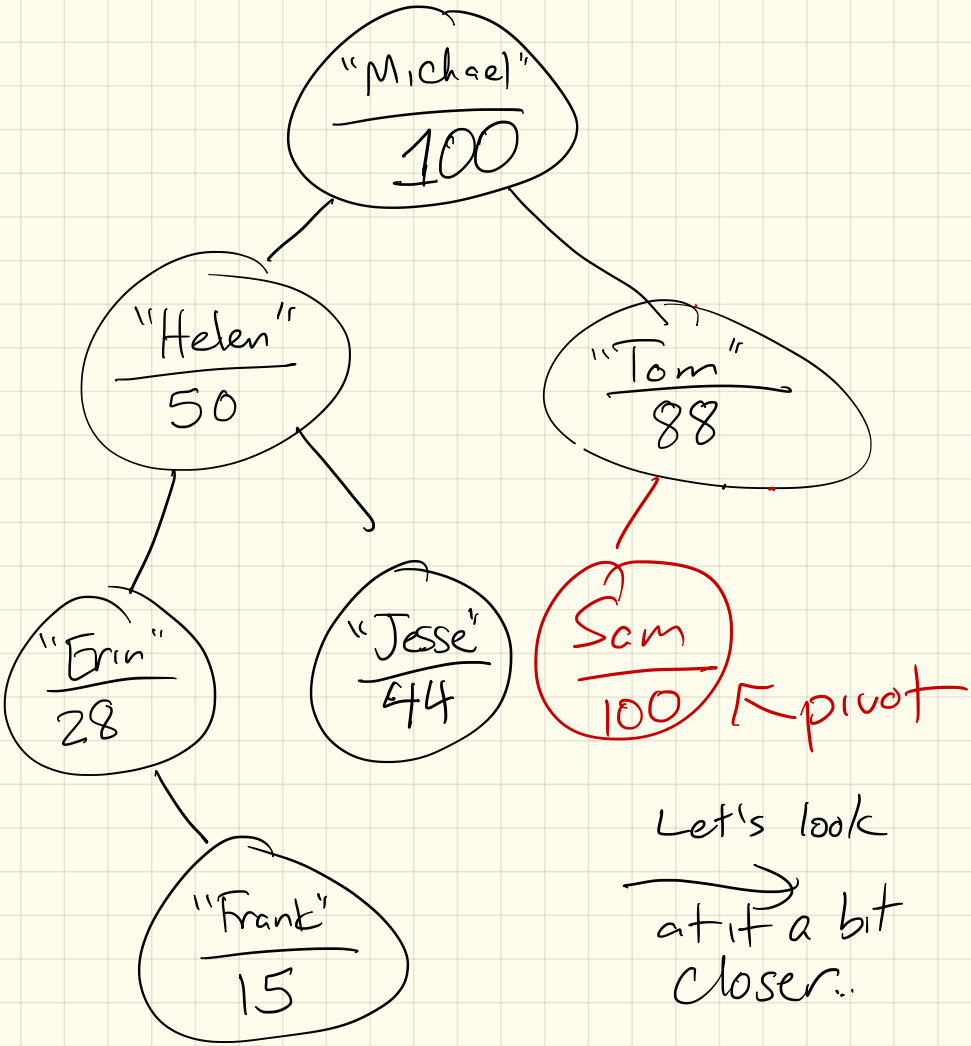
how: insert ("Sam", 101)

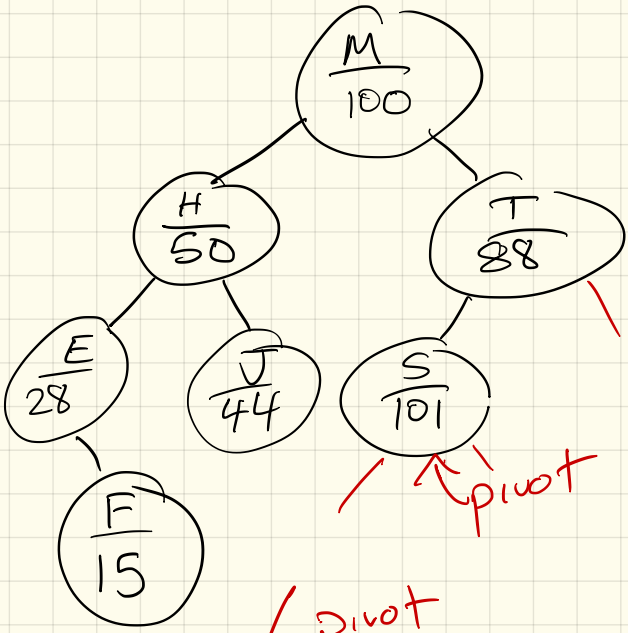
In heaps, we "bubbled" up. Can we do that here?

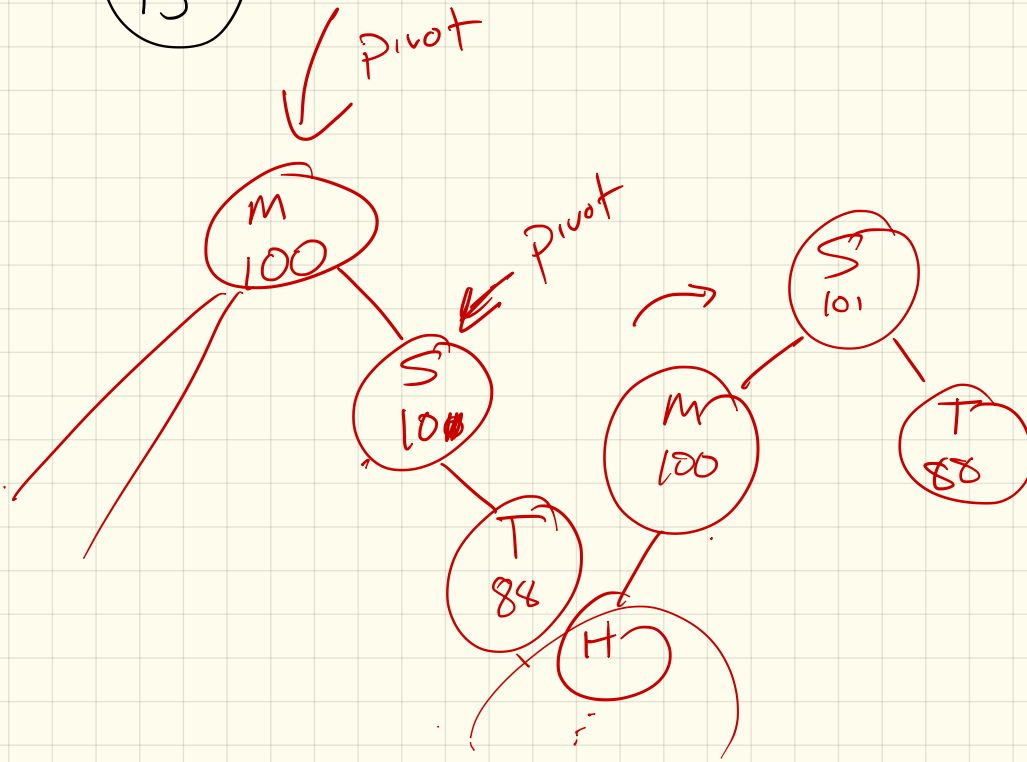


Well - can't violate BST!

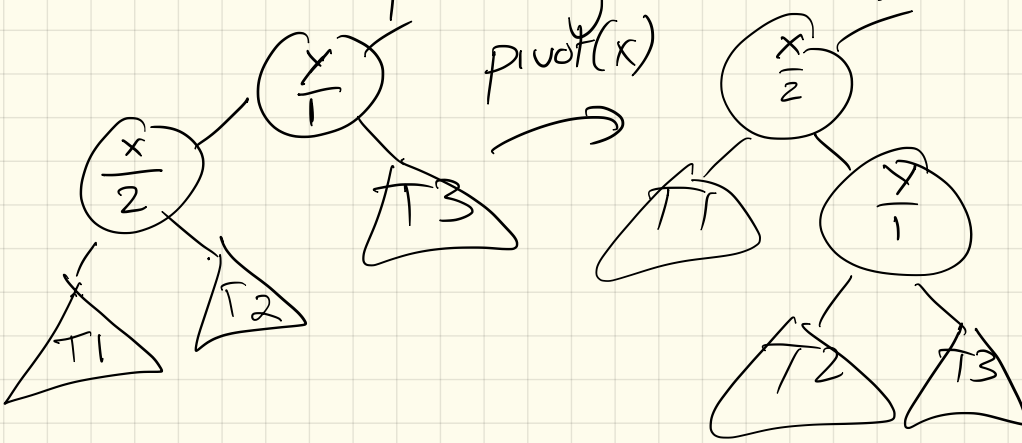
What did we do to move things around in AVL trees?




M
100
H
50
T
88
E
28
J
44
S
101
F
15
pivot


pivot
M
100
pivot
S
101
S
100
M
100
T
88
T
84
H

Result of pivoting: insert(x, 2)

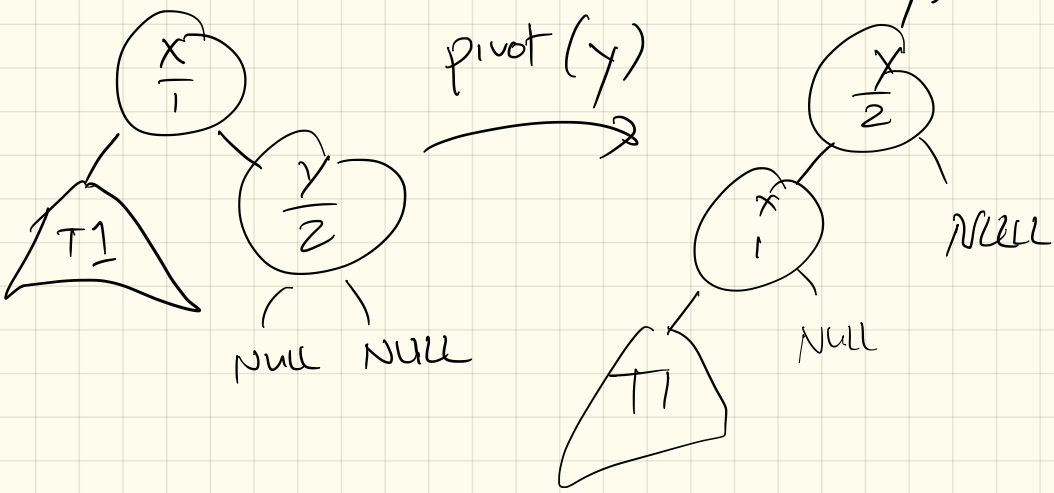


Clearly, we're still a BST!

Can T1, T2, or T3 be non-heaps after this?

(Note: just inserted x, so what are T1 + T2?)

Same for other case: insert(x, 2)



Result:

Insert (val, key):

Run BST insert

Save its location, it

while ((it.priority > it's parent's priority))

pivot(it)

Implementation:

- Inherit from binary search tree

  _data:

  _aux:

- use BST's insert/remove, + binary tree's pivot to fix

Note: Treaps are unique!

Given a set of values/keys, order of insertion is irrelevant.

pf: Consider one valid treap w/ set of values + keys.

Consider x, a node.

If we change x's height:

If we change x's order:

So: Construct treap from following insertions:

(A, 16), (C, 11), (Q, 50), (Z, 41), (M, 30), (L, 2), (S, 55)

What is the root?

Randomized treaps:

Alternative to AVL trees.

Given a value to insert, give it a random priority.

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

Why? remember quicksort: