CS180 - Treaps

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.

\[
\text{tree heap = treap}
\]
Example:

Letters - data (BST)
Numbers - priorities (heap) min heap

```
       M
      /  \
     H    T
    /\    /\  \
   G  F  R  V
  /\  /\  /\  /\  \
 A  6 4  F  5  L  8
```
Insert

Insert: (S, 0)

Problem: violated heap property
In heap we “bubbled up”.
Will that work here?
Here, need BST order.
Rotations

$x \neq y$ are in correct BST order, with $x < y$, but priorities are wrong.

Fix:

```
  x
  |
  y
  |
  x
  |
  y
```

(Insert $x, 0$)
So: insert \((S, 0)\)
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 intertwined)
Draw tree with \((A, 4), (C, 2)\), \((x, 11), (M, 3)\), \((x, 11)\), \((z, 5)\)

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 = pivot