CS180 - Trees

Announcements

- Next HW - due next Thursday
- Midterm 2 - a week from Monday (no lab next week)
- Next HW - after midterm
Next HW: remove in AVL

remove (88)

pivot(y)
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.
Example:
Insert: \((5, 0)\)

Diagram: (Tree structure with nodes labeled with fractions and placeholders for Insert operations.)
In heap, we "bubbled up". Will that work here?

\[
\begin{array}{c}
\frac{m}{4} \\
\frac{n}{2} & & \frac{t}{3} \\
\frac{g}{7} & \frac{f}{4} & \frac{r}{5} \\
\frac{a}{9} & \frac{e}{8} & \frac{s}{6}
\end{array}
\]

Violates BST property. Pivot!
Rotations

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

Fix:

```
    x
   / \
  y  1/2
 /     \
T1      T3
```

(pivot)
So: insert (S, 0) (See prev. slides)
Downside: What can height be? Can we force them to be balanced?

$O(n)$
Treaps are unique!
order of insertion doesn't matter
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 $O(\log n)$. 
Code: How do we implement?

- Change node class
- already coded pivot

() Interest to new class
- redo find, insert, remove