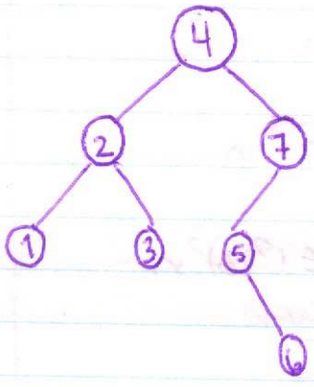
Scapegoat ~~Spacegoat~~ Trees

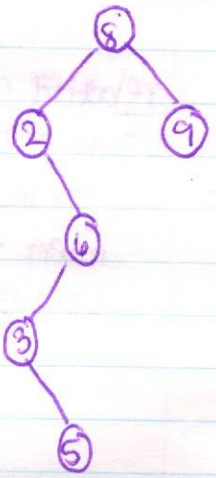
NOV 10
COMP 2402



$O(\log n) \leq \log_{3/2} q$

n = # of nodes
q = controller for size
↳ $q = n$ initially

$n \leq q \leq 2n$
height $\leq \log_{3/2} q$
$= \log_{3/2}(n) + \log_{3/2}(2)$
$= O(\log n) + O(1)$

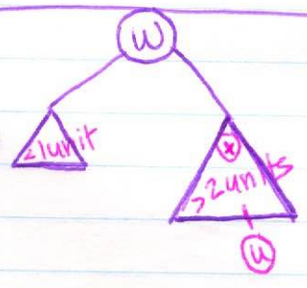



```
remove(x):
  BST.remove(x)   // O(log(n))
  n--;
  if (q > 2n)
    rebuild(root)  // O(n)
    q=n
```

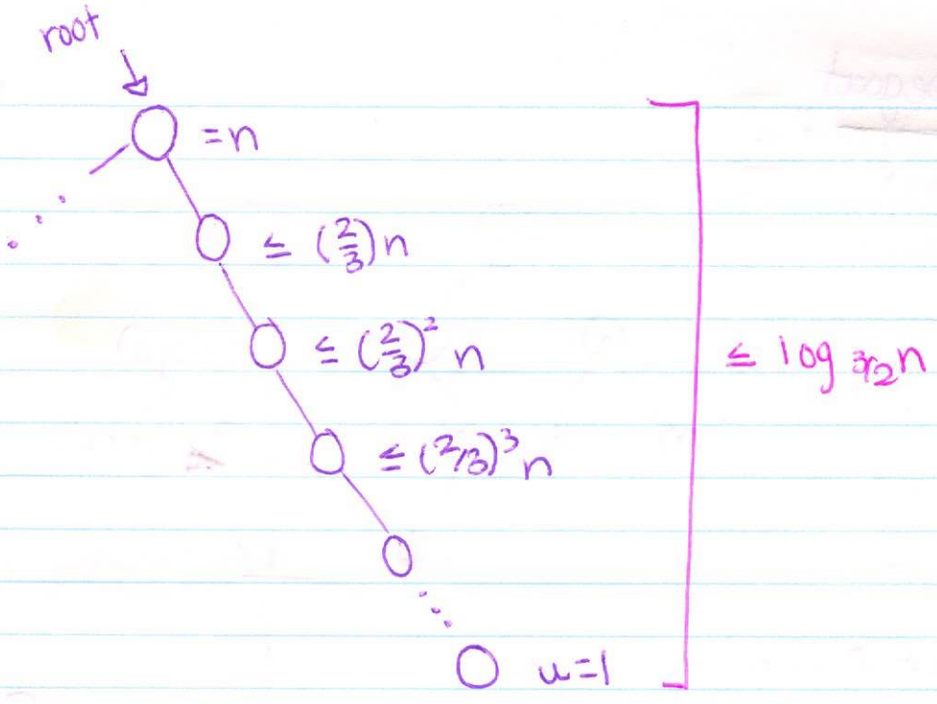
$n = 6$
$q = 6$
$h = 4$
$\log_{3/2}(q) = 4.4 \geq h$

```
add(x)
  BST.add(x)  // O(log(n))
  n++
  q++
  if (depth(u) > log_{3/2} q)
    w = findscapegoat(u)
    rebuild(w)  // O(n)
```

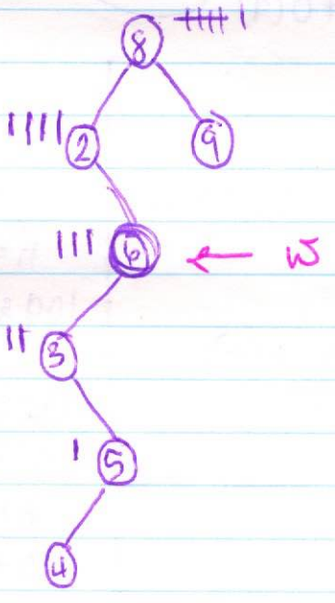
w is a scapegoat
if size(x)
$> \frac{2}{3}$ size(w)



Hilroy



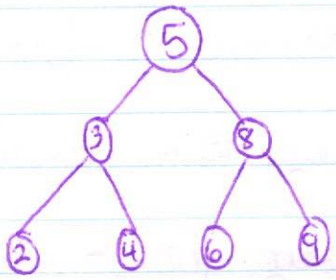
rebuild (w)
in order traversal @ w
↳ build a sorted array
insert the median element @ root
↳ repeat on left/right subtrees. // O(size(w))



then: credits = 0
after last rebuild @ w
imbalance ≤ 1

Now: credits = 1
imbalance = |left - right|
≤ |2/3 size(w) - 1/3 size(w)|
= |1/3 size(w)|

[2, 3, 4, 5, 6, 8, 9]



Then $q_0 = n_0$

after last rebuild

credits jar = 0

Now $q > 2n_1 \rightarrow q - n > n_1$

credits jar $\geq \Delta n$

$= |n_0 - n_1|$

$= q - n_1$