Aumenting Data Structures

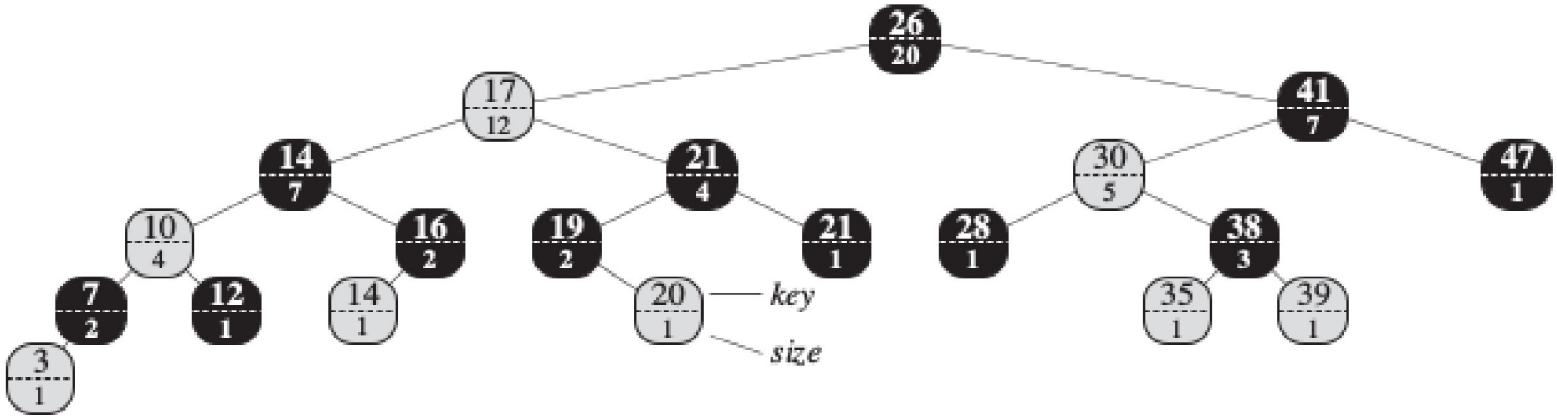
Chris Chávez

Universidad Nacional de San Agustín - Escuela Profesional de Ciencia de la Computación

May 22, 2016

Order Statistics Tree

- The i^{th} order statistic of a set of n elements, where $i \epsilon \{1, 2, ..., n\}$, is simply the element in the set with the i^{th} smallest key.
- Compute the **rank** of an element in $O(lgn)$ time.


26
20
17
12
41
7
14
7
21
4
30
5
47
1
10
4
16
2
19
2
21
1
28
1
38
3
7
2
12
1
14
1
20
1
35
1
39
1
3
1
key
size

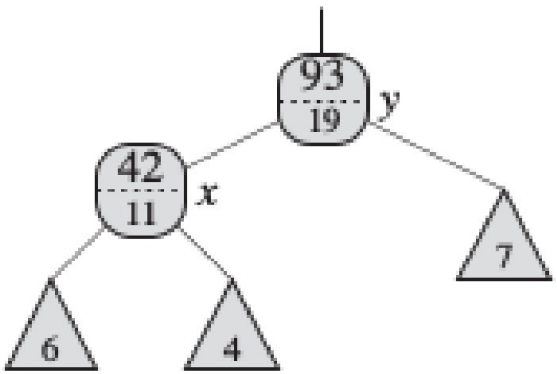
```
OS-SELECT(x, i)
1  r = x.left.size + 1
2  if i == r
3        return x
4  elseif i < r
5        return OS-SELECT(x.left, i)
6  else return OS-SELECT(x.right, i − r)
```

```
OS-RANK(T, x)
1  r = x.left.size + 1
2  y = x
3  while y ≠ T.root
4       if y == y.p.right
5            r = r + y.p.left.size + 1
6       y = y.p
7  return r
```

Maintaining subtree sizes

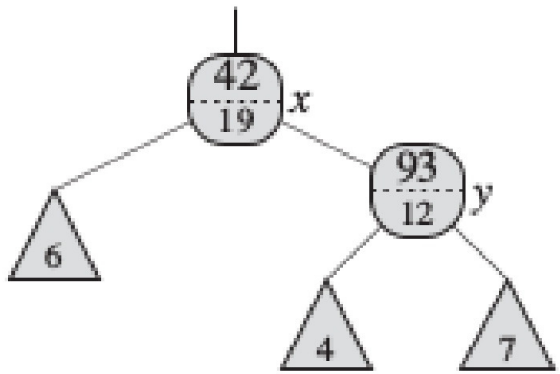
- **Step 1:** We simply increment $x.size$ for each node x on the simple path traversed from the root down toward the leaves.
- **Step 2:**

$$
\begin{aligned}
13 \quad & y.size = x.size \\
14 \quad & x.size = x.left.size + x.right.size + 1
\end{aligned}
$$



LEFT-ROTATE(T, x)

RIGHT-ROTATE(T, y)



How to augment a data structure

1. Choose an underlying data structure.
2. Determine additional information to maintain int hte underlying data structure.
3. Verify that we can maintain the additional information for the basic modifying operations on the underlying data structure.
4. Develop new operations.