Lecture-09: More on BSTs and Midterm #1 (Hand Out)

⊙ how to add(y)?

→ use find like procedure to see if y is already in tree

→ if not, let v be the last node searched. (node v has atmost 1 child)

→ if $y < v.x$, make new node for y and set as left child of x.

→ if $y > v.x$, make new node for y and set y as right child of x

• how long does this take?

– $O\binom{\text{depth}}{\text{of tree}} = O\binom{\text{height}}{\text{of root}}$

⊙ How to remove(y)?

→ first find node v storing y

→ 3 cases: (1) v is leaf

(2) v has one child

(3) v has 2 children

Case 1: If $v == v.parent.left$, set $v.parent.left = null$

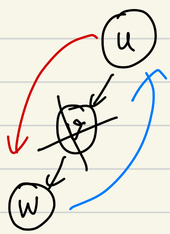
If $v == v.parent.right$, set $v.parent.right = null$

Case 2 (1 child): Assume $v.left \neq null$, $v.right == null$

set $v.left.parent = v.parent$ [sim. other case]

Assume $v == v.parent.left$ [sim. other case]

set $v.parent.left = v.left$



Case 3 on next page

Case 3 (2 children)

choice: replace the value of v with smallest ~~value~~ value larger than v.

→ What node w stores this value?

- go to right child of v
- go to left decendants until no left left.
  → this node is w.
- copy the value w to v ($v.x = w.x$)
- remove the node w (at most 1 child) → either case 1/2.

- how long does this take?
  - $O\begin{pmatrix}\text{depth} \\ \text{of tree}\end{pmatrix} = O\begin{pmatrix}\text{height} \\ \text{of root}\end{pmatrix}$

Questions:

1. how many element/nodes can a binary tree of height h store? $(2^{n+1} - 1)$

2. Min Height of BT with nodes? $(\log(n+1) - 1)$

3. What is max height of BT with n-nodes? $(n-1)$