CS2100 - AVL Trees

Announcements
- Scholarship deadline next week
Remove: cases

15

If (it. hasLeftChild)
  remove (12)
  if (it. hasRightChild)
    remove node

11

4

10

20

25

remove a leaf - easy!
Case 2: \textbf{remove(15)}

1. Null
2. \textbf{17}
3. \textbf{16}
4. \textbf{19}
5. \textbf{20}
6. \textbf{23}
7. \textbf{29}
8. \textbf{17}
9. \textbf{16}
10. \textbf{19}
11. \textbf{23}
12. \textbf{29}

Delete and promote right child.
Case 3: 2 children: find next node in an inorder traversal, delete (20) (in sorted order)

\[ \text{if } \text{it} < \text{20} \text{, go left} \]
\[ \text{if } \text{it} > \text{20} \text{, go right} \]
\[ \text{if } \text{it} = \text{20} \text{, copy } \text{it} + 2 \text{ into } \text{it}, \text{ delete } \text{it} \text{ and promote right child of it} \]

\( i = i + 1 \)
Recap: BST

Runtimes:

- Insert \( \Rightarrow \) worst case, need to travel from root to some leaf \( \Theta(\text{height}) \)
- Remove
- Find

Here: \( O(n) \)
Consider this tree:

Take out a piece of paper.
Redraw & make this as “good” as possible.
What did you do?
Balanced Binary Search Tree

- Red-black trees \(\leq 1.2 \log n\)
- Splay Trees
- AVL trees \(\leq 1.4 \log n\)

Goal of all: \(O(\log n)\)
AVL Trees

Height - Balance Property:
For every node of T, the heights of the children differ by at most 1.

=> max height =

(How do we calculate height again?)
Ex:
Now: How can we mess this up?

(In other words, how can the height change?)
Insert:

\[ \text{insert}(54) \]
So: consider the lowest node which does not satisfy the height-balance property. Call this $z$.

Let $y$ be $z$'s child with larger height.

Let $x$ be $y$'s child with larger height.

Now - fix it!
What did you do?
Another - insert 49
So consider the lowest node which does not satisfy height-balance property \( \varnothing \) - call this \( \tau \).

Let \( x \) be \( \tau \)'s child with larger height.

Let \( y \) be \( y \)'s child with larger height.

Now - fix it!
What did you do?
Generalize — Consider $x, y, z$. How can we restructure? (Hint: What is in-order traversal of these in each case?)
Actual picture:

Where do the sub-trees go??
Another
Any way you do this, “2” becomes the root of the new subtree with “1” to the left and “3” to the right!

What about $T_1, T_2, T_3, \text{ and } T_4$?
Key operation: Pivot