

Today - Lecture 15 - CS163

- 1) Review 2-3, 2-3-4, Red Black and AVL
- 2) Topic 10 - B-Trees (slide 10-9)
- 3) Topic 11 - Heaps
- 4) Next - Preview of the deletion algs!

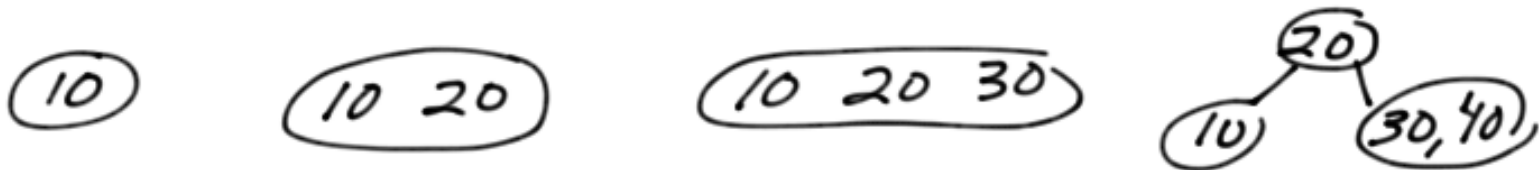
Announcements:

- Practice Recursion!
- Practice BST code!
- Practice Advanced Trees!

Review - 2-3-4 trees

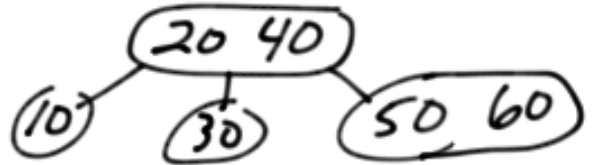
What if data is inserted in sorted order?

10 20 30 40 50 60 70 80 90



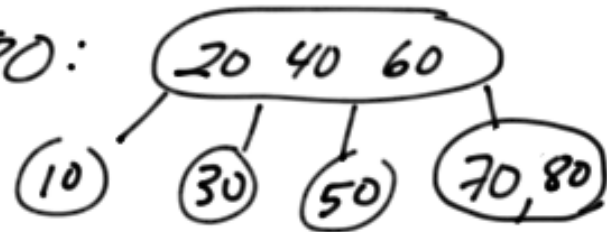
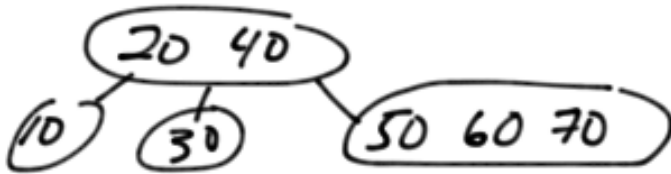
Add 50:

Add 60:



Add 70:

Add 80:

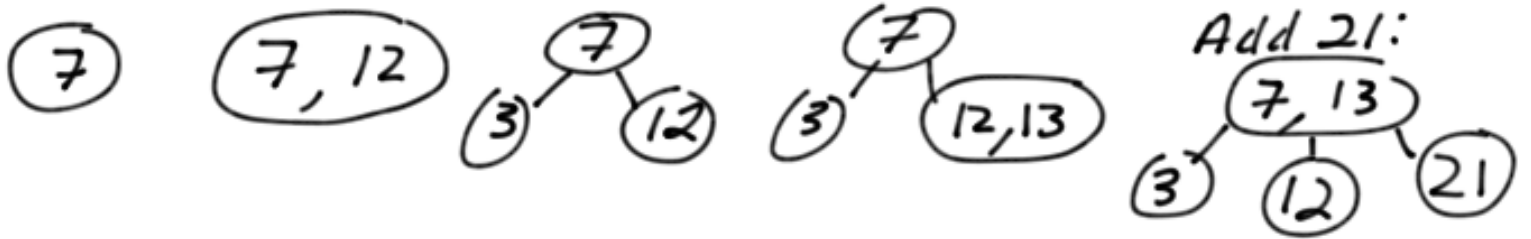


Add 90:

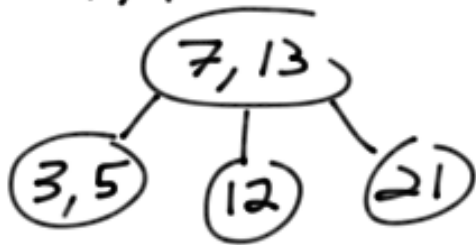


Practice: Build 2-3 Tree

7 12 3 13 21 5 8 50



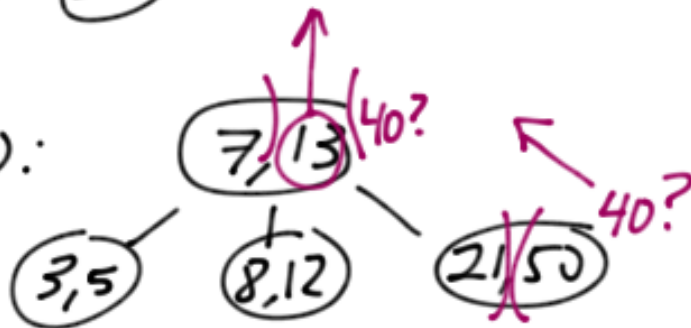
Add 5:



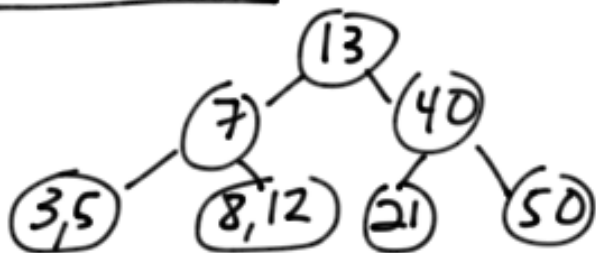
Add 8:



Add 50:

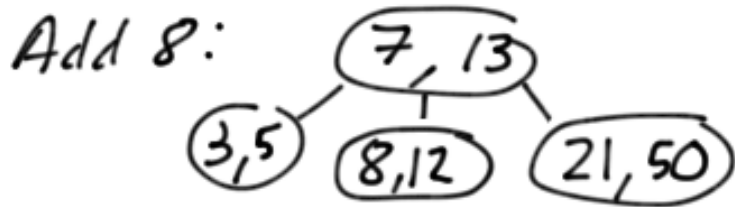
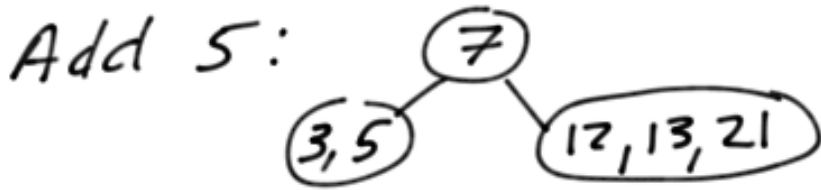
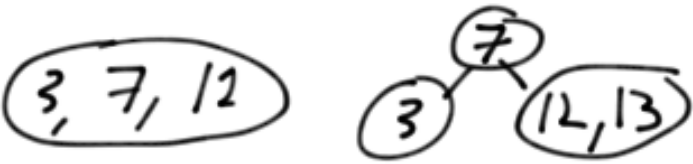


Add 40:



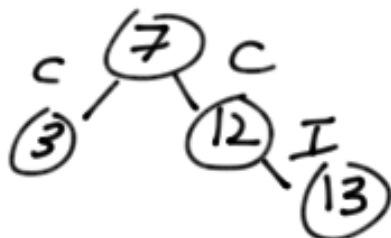
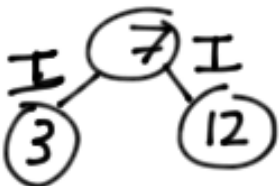
Practice: Build a 2-3-4 Tree

7 12 3 13 21 5 8 50



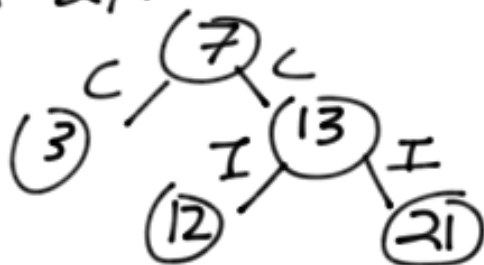
Practice: Build a Red-Black Tree

7 12 3 13 21 5 8 50



I - Internal
C - child

Add 21:

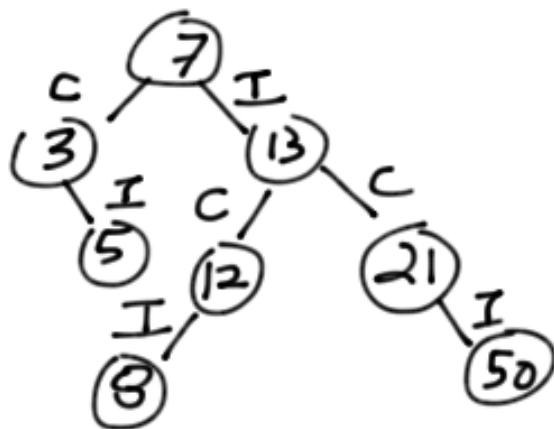


Add 5:



(Red) →

Add 8:



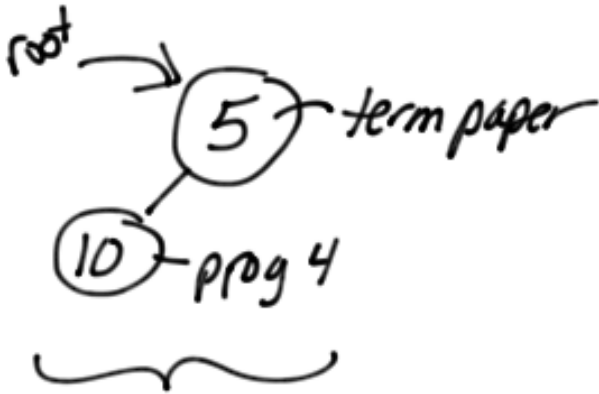
ADD 50

Practice - Build an AVL Tree

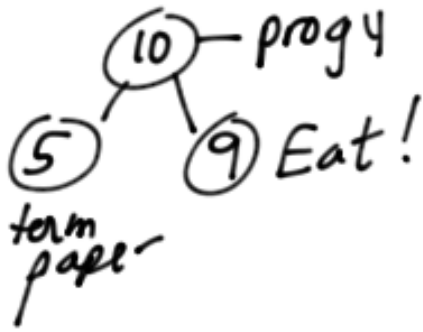
7 12 3 13 21 5 8 50

Heap-insert

10 - High priority
1 - low priority

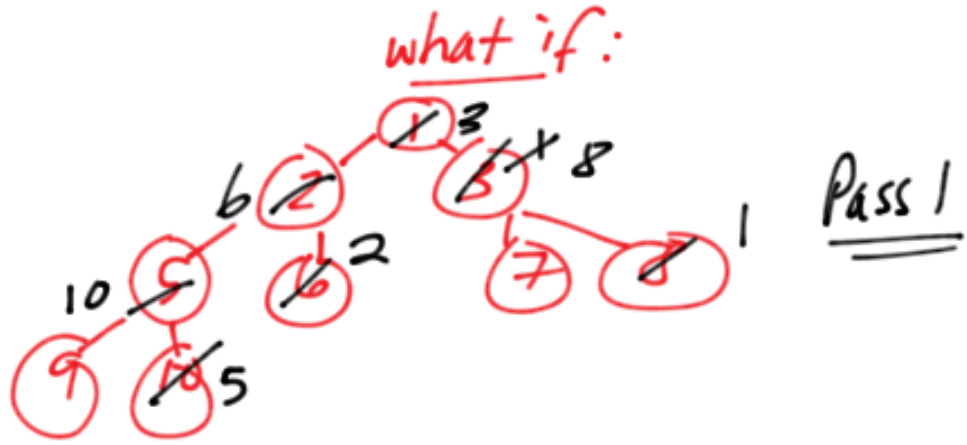


Add 3 7 6 4



5 10 9 3 7 6 4

Heap Sort

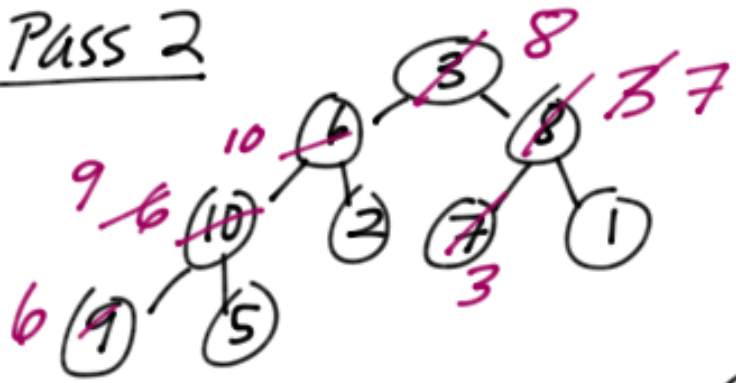


5 10 9 3 7 6 4

Heap Sort



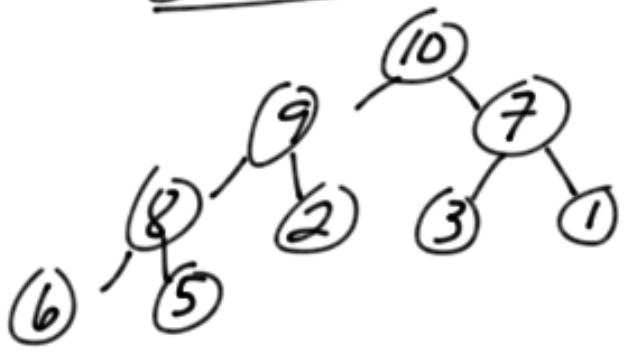
Pass 2



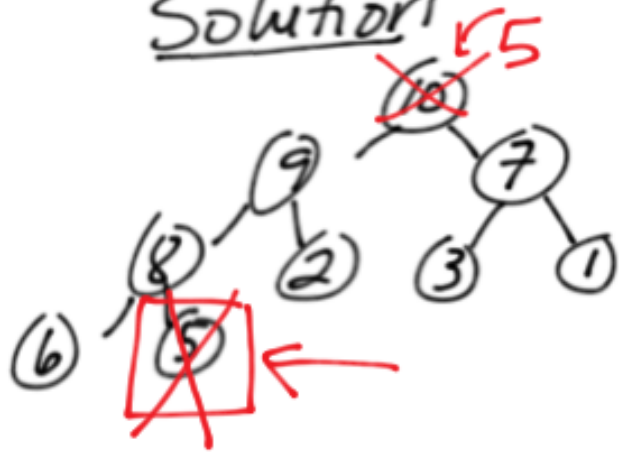
Pass 3



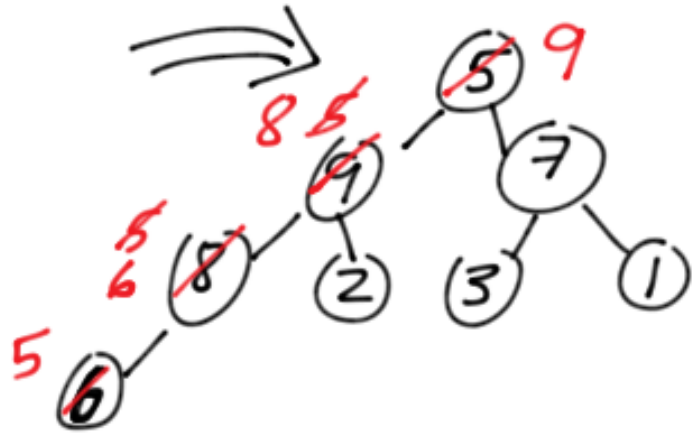
Solution



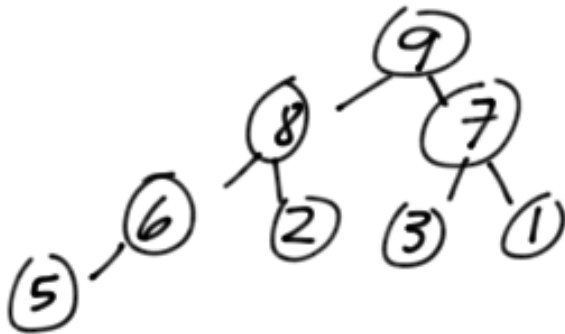
Solution



Remove



Solution After Removing 10



Practice BST code -

- 1) Write the code to copy every **leaf** of a BST and place it into a LLL
- 2) write the code to make a copy of a BST and place it into an Empty LLL
- 3) Write the code to display the **LARGEST** item in a BST
- 4) write the code to remove the **LARGEST** from a BST
- 5) write the code to count the # of items in a BST
- 6) Revise #5 for a 2-3 tree
 - count just the # of nodes
 - or, count **all** data!

