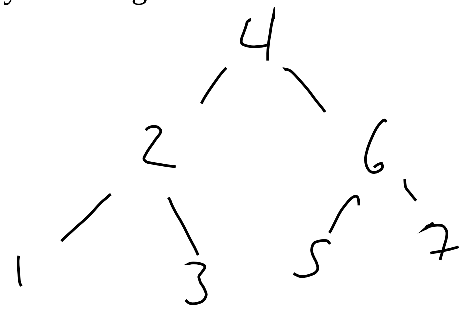
CSE332 15su 2015-07-16

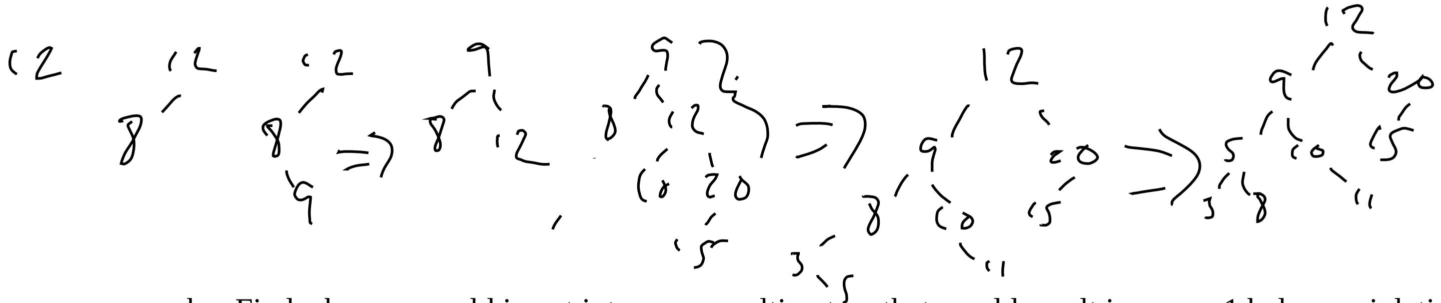
Section 4 Worksheet: AVL Trees & B-Trees

1. What 3 properties must an AVL tree have?
   a. Binary Tree, $|\text{children}| \le 2 \ \forall n$
   b. Binary Search Tree $n.\text{left}.\text{val} \le n.\text{val} \le n.\text{right}.\text{val}$
   c. Balanced $|n.\text{left}.\text{height} - n.\text{right}.\text{height}| \le 1 \ \forall \text{nodes } n$

2. In a typical BST, inserting keys in order result in a worst-case height. Show the result when an initially empty AVL tree has keys 1 through 7 inserted in order.



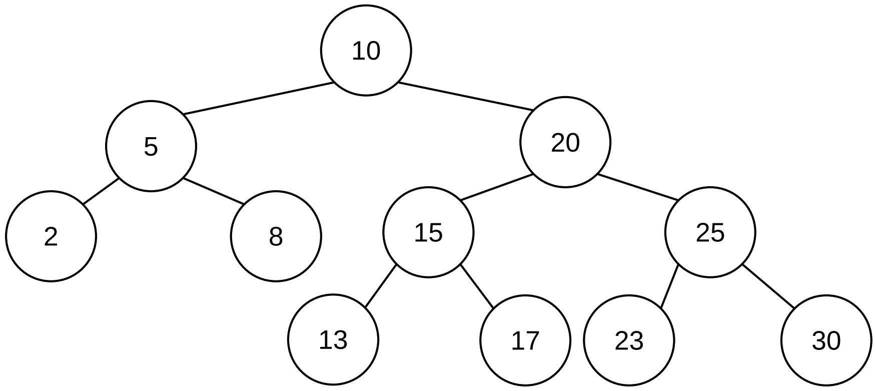
3. AVL tree balance violation cases:
   a. Insert the following keys, in order, into an initially empty AVL tree: 12, 8, 9, 20, 10, 15, 3, 11, 5.



   b. Find a key we could insert into your resulting tree that would result in a case 1 balance violation (left-left).

any value $x < 3$ or $12 < x < 15$

4. For the following AVL tree:



   a. What values could you insert to cause a right-right imbalance, and at which node does the imbalance occur?

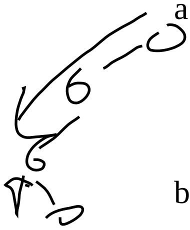
   b. How about a right-left imbalance? At which node does the imbalance occur?

   c. Insert 18 into the following AVL tree. What type of imbalance does it cause? Show the result after balancing.

5. Given a binary search tree, describe how you could convert it into an AVL tree with worst-case time O(nlogn) and best case O(n).

6. Say you work for a hospital where patients are seen in the ER based on a priority level assigned by the triage nurse. Your boss wants you to implement a data structure that stores the priority level and patient record, and will display who the next patient to be seen is. He suggests you just use the AVL tree you already have implemented, since you findMin/insert/deleteMin all run in O(logn) time guaranteed anyway.

   a. Why might you want to use a binary min heap instead?



      min @ top → O(1)

      insert faster amortized O(1)

   b. What kind of situation might your AVL tree be more useful for at the hospital?

      print in-order traversal

7. B-Trees:

   a. What constraints do the following values impose on a B-Tree: M=32, L=16?

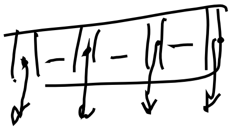
   b. Insert the following into an empty B tree with M=3 and L=3: 12, 24, 36, 17, 18, 5, 22, 20.

   c. Delete 17, 12, 22, 5 & 36

8. Given the following parameters for a B-tree with M= 11 and L = 8
Key Size = 10 bytes
Pointer Size = 2 bytes
Data Size = 16 bytes per record (includes the key)

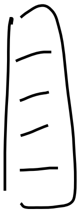
Assuming that M and L were chosen appropriately, what is the likely size of a disk block on the machine where this implementation will be deployed? Give a numeric answer and a short justification based on two equations using the parameter values above.

Internal Node
$M=11$



$\text{key size} \cdot (m-1) +$
$\text{pointer size} \cdot m$
$10 \cdot 10 + 2 \cdot 11 = 122\ B$

Leaf
$L=8$



$\text{data} \cdot L$
$8 \cdot 16 = 128\ B$

$128 = 2^7\ B$