(these review questions are not to be answered in writing in class. At some point, you should make sure you can answer them).

How can we show that the maximum height of an AVL tree with n nodes is Θ(log n)?

What extra data must we store in each AVL tree node in order to efficiently keep the tree balanced?

Give a high-level description of the algorithm for ensuring that an AVL tree is balanced after we do an insertion. You do not need to describe the details of particular rotations.

What is the maximum number of rotations required to rebalance an AVL tree after an insertion? Circle one

Θ(1) Θ(log n) Θ(n) Θ(n log n) Θ(n^2)

If a rotation is required after insertion, which of the following applies to the height of the tree after the insertion and rotation? Circle one.

a. Height is always the same as before the insertion.
b. Height is usually the same as before the insertion, but may be larger.
c. Height is usually the same as before the insertion, but may be smaller.

Write a Θ(log N) function which, given a pointer to the root of an AVL tree, returns the height of the tree.

2. Review a proof that the maximum height of a height-balanced tree with N nodes is Θ (log N):

3. What property do all of the leaves in a 2-3 tree have?

4. What are the upper and lower bounds on the height of a 2-3 tree with n elements?

5. Continue the following sequence of 2-3 tree insertions by adding 10, 11, 12, …

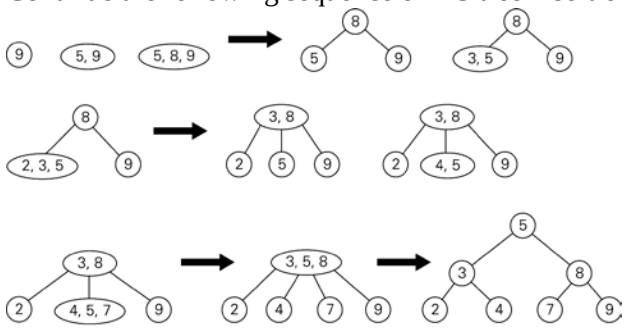
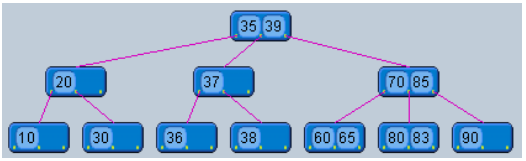


FIGURE 6.8 Construction of a 2-3 tree for the list 9, 5, 8, 3, 2, 4, 7

Show the result of inserting 84 into this tree: 

6. A binary *max heap* is a complete binary tree with the additional property that …

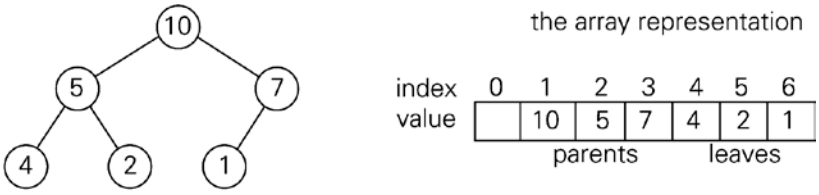


FIGURE 6.10 Heap and its array representation

7. Describe the details of the representation of a binary (max) heap by an array.

   Runtime for `insert` and `removeMax`: (why?)

8. Give a brief overview of how heapsort works.