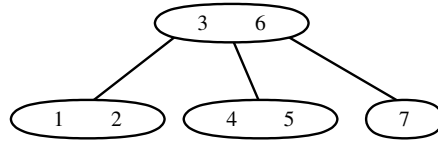
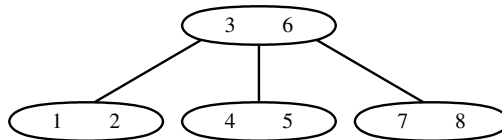

csce350 — Data Structures and Algorithms
Fall 2019 — Extra Notes on 2-3 Trees

This document is a brief example of the process of inserting into a 2-3 tree.

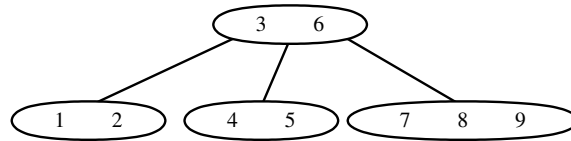
Suppose we want to insert a new key, 8, into this 2-3 tree.



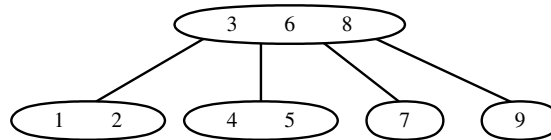
At the root, compare 8 to 3 and 8 to 6 to determine that we need to follow the right child. At the right child, compare 8 to 7 to determine that 8 should be on the right side. Since we're at the leaf level, we can insert it here.



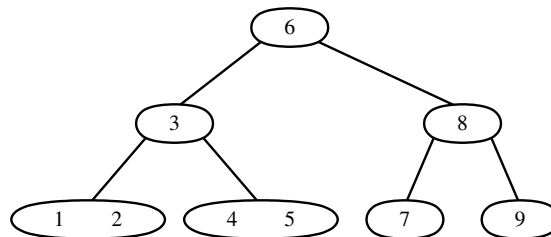
Notice that the node now has 2 keys (7 and 8), which is allowed in a 2-3 tree. So we are done with this insert. Next, suppose we want to insert again, this time a 9. Following the same process, and remembering that we always insert at the existing leaf level, we get this:



But this node now has 3 keys, and we can only have at most two. So we split the 789 node and 'promote' the middle key (8) to the next level up.



Alas, now the root node has too many keys. So we need to repeat the process again. Split the 368 node and promote the middle key, which is 6.



In this case, since we're splitting the root node, we just create a new root containing the promoted key. Notice that we always, even when dealing with this kind of 'overflow' of a node, have one more child than we have keys.