Exercises on Trees

1. Suppose you have 2000 nodes that you need to put into a binary search tree. What are the maximum and minimum possible heights of the tree?

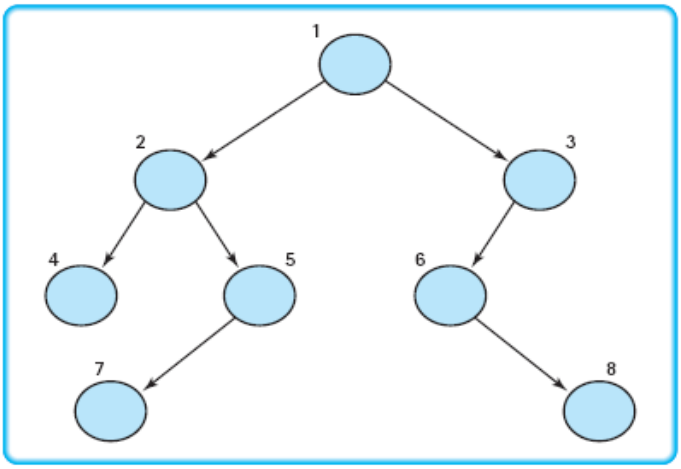
2. If a binary tree has N levels (assume root is at level 0), what is the maximum and minimum possible number of nodes in the tree?

3. If you wanted to traverse a binary search tree, writing all the elements to a file, and later (the next time you ran the program) rebuild the tree by reading and inserting, would an in-order traversal be appropriate? Why or why not?

4. Draw the binary search tree whose data elements are inserted in the following order:
50 72 96 94 107 26 12 11 9 2 10 25 51 16 17 95
(comparison here is integer comparison between data elements)

5. In the tree above (#4.), show the effect of each of the following operations:
delete(25)
delete(94)
delete(50)
Consider
(i) Each operation carried out independently on the original tree and
(ii) Each operation done successively, one after another.

Examine the following binary search tree and answer the questions in Exercises 6-9. The numbers on the nodes are labels so that we can talk about the nodes; they are not key values within the nodes.



6. Draw the tree after deleting node 1
7. Draw the tree after deleting node 5
8. Draw the tree after deleting node 6
9. In the previous three questions, you can replace the deleted node by either its successor or its predecessor. In each of the above three questions provide a second alternative to the deletion operation.

10. Write a member function *void delete(BTree* Node)* in the BTree class to delete a given Node. You may use the code to find successor of a node. Test your function with a few examples.