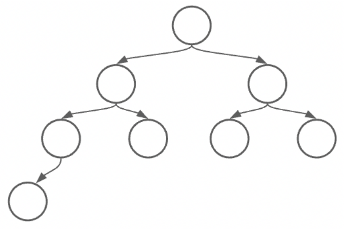
1) Define and draw how a binary tree for the given input sequence looks like:
   Input Integers: 9, 8, 7, 6, 5, 4, 3, 2, 1

   Explanation:
   You will build the tree starting from the left, meaning 9 would be your first node. Then you insert the next number and follow binary tree insertion rule:
   Tree insertion rule: left node < root node < right node
   We can assume we have no duplicate node values

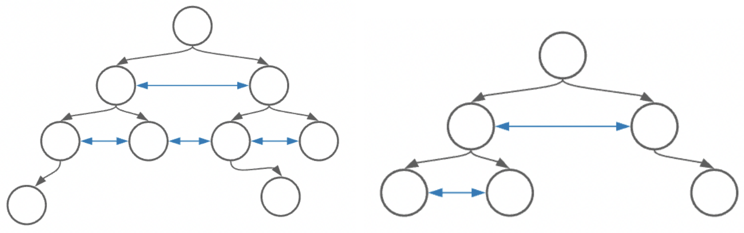
2) Define the tree data structure, with the tree node values being integers
3) Define an algorithm in language of your choice that can create a more balanced tree over the given input defined in #1

   Explanation:
   A balanced tree would look more like below.
   To note, there are Red/Black or similar algorithms which do re-balancing of the tree while inserting into the tree. This would be rather difficult to implement. Here we are looking for something simpler than rebalancing while inserting which is possible due to how the input sequence is defined.
   Example balance tree:



4) Extend the data structure defined in #2 that allows you to navigate bi-directionally on each level of the tree.

   Example tree with level navigation where bi-directional links are added in blue.
   To note, the lowest leaf nodes (left example) are not connected, because only nodes that have a nearest possible neighbor will be connected. Similar on the right example where the lowest leaf nodes are not fully connected due to the same reason.



5) Outline an approach how the bi-directional links can be added while you are inserting the nodes into the tree.