**UNIQUE BT’S**

Consider a binary tree with n nodes from 1,2, 3, n the task is to find the numbers of unique BST’s that can be constructed by changing the insertion order.

If there are n nodes in a BST starting from then, for example we will consider n=3.

Then the number of possible unique BST’s will be 5. This is how we arrive at the result.

* If 1 is the root then there is no element to the left side so we will have f (2) ways
* If 2 is the root then there is one element to the left side and one to the right so we will have

f (1) \* f (1) ways

* If 3 is the root then there is no element to the right side so we will have f (2) ways

So hence concluding and summing up the values

Total possibilities = f (2) +f (1) \*f (1) +f (2)

= 2+1\*1+2

=2+1+2

=5

**CHECK INORDER FOR A GIVEN BST**

The solution is itself straight forward the in order traversal of a BST is always sorted so all we have to do is to check whether the previous number is smaller than the current number.

The basic operation is to compare

**SHORTEST DISTANCE BETWEEN TWO NODES IN AN INFINITE BST:**

A very simple approach with an O(n) complexity is possible as the Tree has a continuous range of numbers each value in a level will the half of any value of the previous level

1= 1  
2= 2 3  
3= 4 5 6 7

here you can see a pattern evolving so to find the distance between the two nodes just try and make the values equal... and count the number of iterations