Problem Statement –

**Perform a breadth first search in a Binary Tree assuming there are no cycles a in the tree.**

Solution 1 –

1. Create 2 Queues. TreeNodesQ, CurrentLevelQ.
2. Read the root value and insert it into the TreeNodesQ. Insert the current level (0) into the CurrentLevelQ.
3. While(The TreeNodesQ is not empty)
   1. Dequeue from TreeNodesQ and Print the Value. Dequeue From CurrentLevelQ to know its level.
   2. Search the left and right nodes of the TreeNode that has been Dequeued. Add the two values to TreeNodesQ. Insert its current level by added 1 to the value which you got from 3(a) into CurrentLevelQ.
   3. Create a new Line when the value of the current Level changes.

**Space Complexity – O(N)**

**Time Complexity – O(1)**

**Average Case – O(N)**

Useful notes (If any) -

1. Now what if there are loops in a tree? Use a hash map to maintain which tree nodes you have visited.
2. One problem with this approach is that you have to maintain 2 Queues. The Second Q is only needed if each level needs to be printed in a separate Line.

Solution 2 –

**Space Complexity –**

**Time Complexity –**

Useful Notes –

This method does not use 2 Queues. Though there isn’t a “Real Improvement” When the solution scales, this is a minior optimization.