Bottom View of Binary Tree GFG

Given a binary tree, print the bottom view from left to right.
A node is included in bottom view if it can be seen when we look at the tree from bottom.

Example:

```
            5
          /    \
        3       7
      /   \      \
   11      1       6
  /   \          /   \
9    12       13    15
```

Output: The Bottom View of Binary Tree: 9, 11, 12, 1, 13, 6, 15

Approach 1: Function to print Bottom View of a binary tree using a iterative approach.

- The iterative approach uses level-order traversal (BFS) to traverse the tree.
- It maintains a map to store nodes at each horizontal distance from the root.
- While traversing, it updates the map with the current node's value for its horizontal distance.
- This way, it ensures that the bottom-most node at each horizontal distance is stored in the map, representing the bottom view.
- Finally, it extracts and prints the nodes from the map in order of increasing horizontal distance.

Time Complexity:

- **The time complexity is O(N), where N is the number of nodes in the tree, as each node is visited once during level-order traversal.**

Space Complexity:

- **The space complexity is O(N) in the worst case, primarily due to the queue used for level-order traversal and the map used to store bottom view nodes.**