





So, we can say that, in the simplest words, the top view of a binary tree refers to the set of nodes visible if we view the tree from an axis parallel to the levels of the tree. Or if we take the topmost nodes of all the vertical orders in it. Vertical orders refer to the set of nodes at a particular horizontal distance.

Though there are various methods to print the top view of a binary tree, we will start with discussing the basic method, which does the [level order traversal](https://www.codingninjas.com/codestudio/library/level-order-traversal-of-a-binary-tree) of the binary tree and prints all the topmost nodes for each value of the horizontal distance. Then move on to advanced methods like two-variable methods etc.

**Basic Approach**

This method does find vertical orders in the tree with the help of the horizontal distance of each node, the horizontal distance of the left child of a node is one less than the parent, and the right child is one more than the parent.

Now we do the [level order traversal](https://www.codingninjas.com/codestudio/library/level-order-traversal-of-a-binary-tree) of the tree to ensure that nodes appear in the top to bottom order for every vertical order, and we maintain a [hashmap](https://www.codingninjas.com/codestudio/library/hashmap" \t "_blank) to keep track of the visited nodes for each vertical order.

**Algorithm for a basic approach**

* Create a binary tree or take it from user input.
* Create a queue to hold nodes during the level order traversal of the tree.
* Create a map to keep track of the visited nodes.
* Start the level order traversal of the binary tree by pushing the root and its horizontal distance into the queue.
* Now run a loop while the queue is not empty-
* Check if the map has a node at that horizontal distance; if not, push it into the map.
* Push the left and right child of the node in the queue, and update the value of their horizontal distance.
* Pop the element from the queue.
* Repeat step 5 at the front of the queue.
* Then print the values on the map.