

Minimum Depth of Binary Tree

Question: Given a binary tree, find its minimum depth.

The minimum depth is the number of nodes along the shortest path from the root node down to the nearest leaf node.

Solutions:

```
class TreeNode:
```

```
    def __init__(self, x):
        self.val = x
        self.left = None
        self.right = None
```

```
class Solution:
```

```
    # @param root, a tree node
    # @return an integer
    def minDepth(self, root):
        if root == None:
            return 0
        if root.left == None:
            return self.minDepth(root.right) + 1
        if root.right == None:
            return self.minDepth(root.left) + 1
        return min(self.minDepth(root.left), self.minDepth(root.right)) + 1
```

```
if __name__ == '__main__':
```

```
    BT, BT.right, BT.right.left, BT.left = TreeNode(1), TreeNode(2), TreeNode(3),
    TreeNode(10)
    print ( Solution().minDepth(BT) )
```