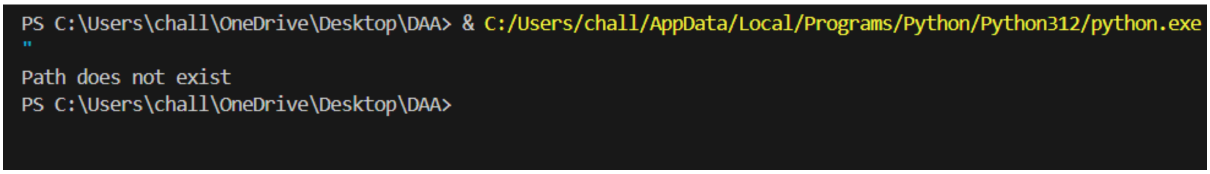
35. Check If a String Is a Valid Sequence from Root to Leaves Path in a Binary Tree Given a binary tree where each path going from the root to any leaf form a valid sequence, check if a given string is a valid sequence in such binary tree. We get the given string from the concatenation of an array of integers arr and the concatenation of all values of the nodes along a path results in a sequence in the given binary tree. Example 1: Input: root = [0,1,0,0,1,0,null,null,1,0,0], arr = [0,1,0,1] Output: true Explanation: The path 0 -> 1 -> 0 -> 1 is a valid sequence (green color in the figure). Other valid sequences are: 0 -> 1 -> 1 - > 0 0 -> 0 -> 0

PROGRAM:

```
class Node:
   def __init__(self, val):
      self.val = val
      self.left = None
      self.right = None
def exist_path(root):
   if root is None:
      return False
      if root.left is None and root.right is None:
         return root.val
   return False
root = Node(5)
root.left = Node(2)
root.right = Node(7)
root.left.left = Node(1)
root.left.right = Node(4)
root.right.left = Node(6)
root.right.right = Node(8)
if exist_path(root):
    print("Path exists")
else:
    print("Path does not exist")
```

OUTPUT:



```
PS C:\Users\chall\OneDrive\Desktop\DAA> & C:/Users/chall/AppData/Local/Programs/Python/Python312/python.exe
"

Path does not exist
PS C:\Users\chall\OneDrive\Desktop\DAA>
```

TIME COMPLEXITY:

Time complexity for the above code is

F(n)=O(n)