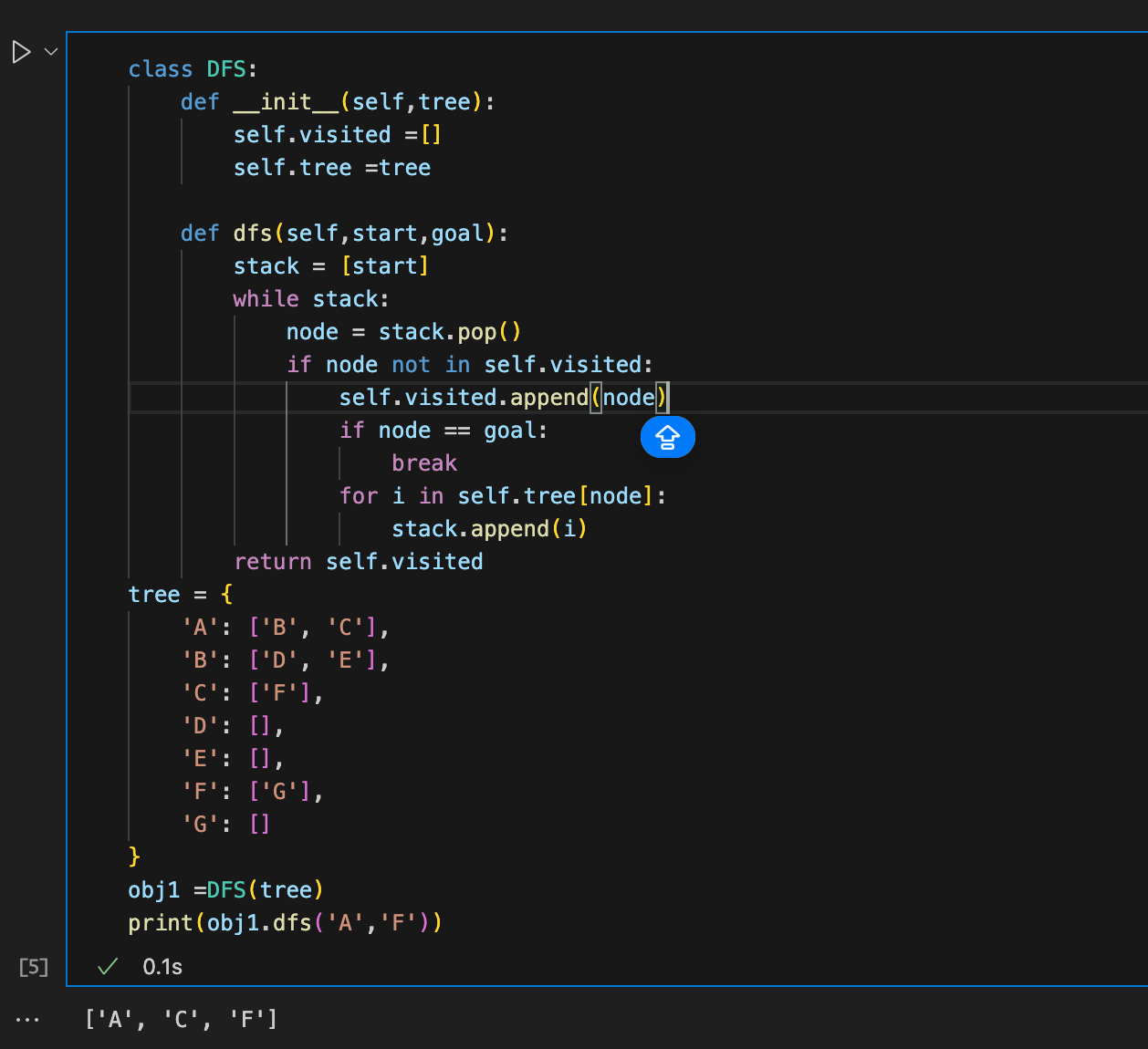
**Lab-task-5**

**Name: Najia Khan(123)**

**Q:1 DFS with Stack**

**Key Points**

* DFS (Depth First Search) explores tree/graph by going deep before backtracking.
* Stack is used (LIFO order).
* In this code:
  + visited stores the nodes in the order they are visited.
  + Loop continues until stack becomes empty or goal is found.
* Example traversal (till goal = F): ['A', 'C', 'F'].
* If goal was not given, DFS would explore whole tree



**Q:2 Research about "Inorder, Preorder, Postorder" and implement in DFS**

* **Preorder:** Visits the root node first, then recursively visits left/right children in order. Output order = Root → Left → Right.
* **Postorder:** Recursively visits children first, then adds the root at the end. Output order = Left → Right → Root.
* **Inorder:** Goes to the left child, then root, then right child. Works naturally for binary trees (like BST). Output order = Left → Root → Right.

👉 In your code, the tree is a dictionary, so:

* preorder('A') will list nodes starting from A before children.
* postorder('A') will list children before their parent.
* inorder('A') handles only 2 children (left/right) properly, not more than 2.

