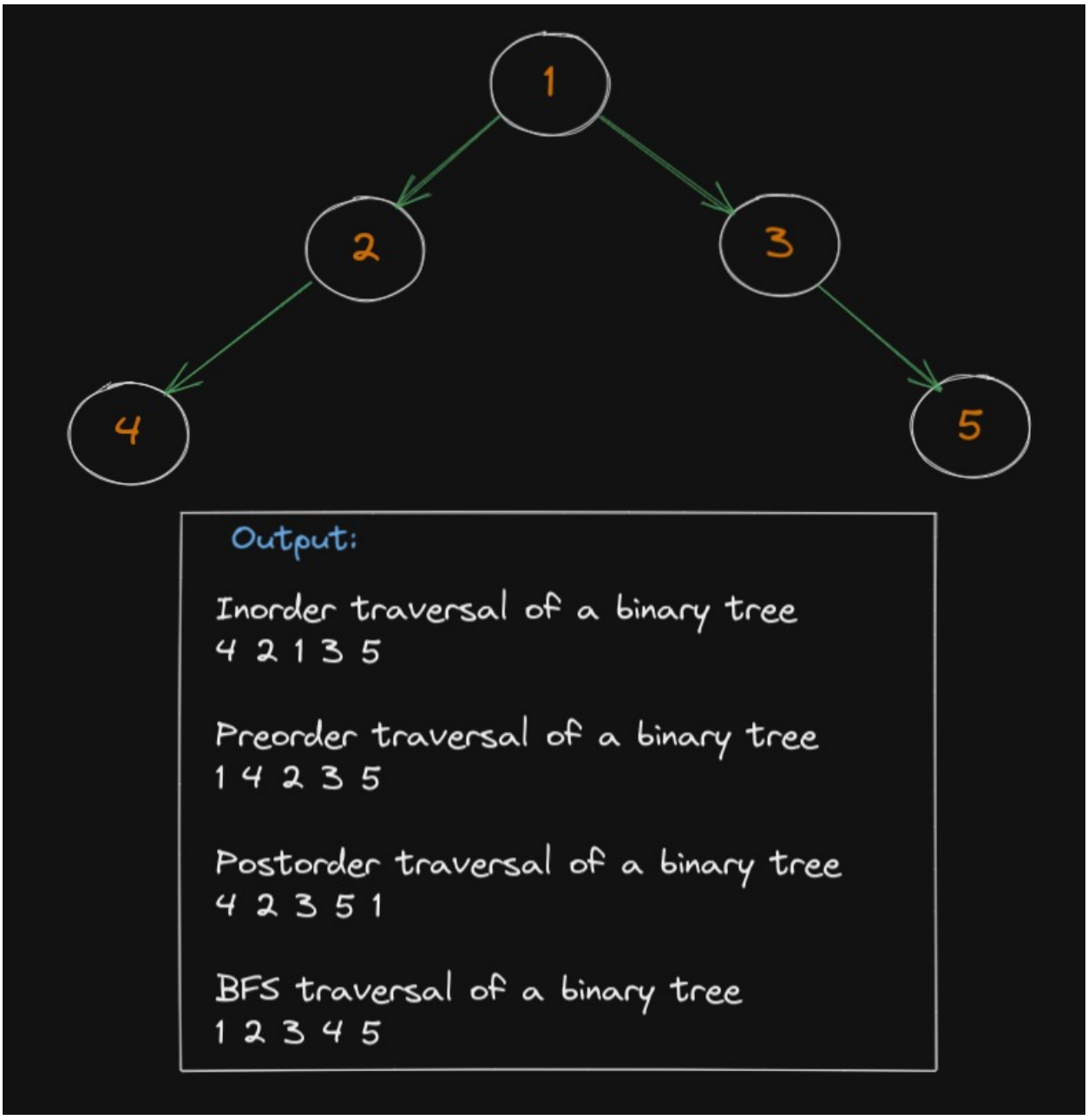
LP-II Assgn-1: DFS & BFS of a tree

Code

```
class Node:
    def __init__(self, value):
        self.left = None
        self.right = None
        self.data = value

def inorder(root):

    if root:
        inorder(root.left)
        print(root.data, end = " ")
        inorder(root.right)

def preorder(root):

    if root:
```

```
        print(root.data, end = " ")
        inorder(root.left)
        inorder(root.right)

def postorder(root):

    if root:
        inorder(root.left)
        inorder(root.right)
        print(root.data, end = " ")

def bfs(root):

    if root is None:
        return

    queue = []
    queue.append(root)

    while (len(queue) > 0):
        print(queue[0].data, end = ' ')
        item = queue.pop(0)

        if item.left is not None:
            queue.append(item.left)

        if item.right is not None:
            queue.append(item.right)

root = Node(1)
root.left = Node(2)
root.right = Node(3)
root.left.left = Node(4)
root.right.right = Node(5)

print('Inorder traversal of a binary tree')
inorder(root)
print(" "), print(" ")

print('Preorder traversal of a binary tree')
preorder(root)
print(" "), print(" ")

print('Postorder traversal of a binary tree')
postorder(root)
print(" "), print(" ")

print('BFS traversal of a binary tree')
bfs(root)
print(" "), print(" ")
```

Tree visualization



Output

```
vedant@vedant-ubuntu:~/VedWORK/PICT/TE/SEM 6/LP-II Lab/assgn1$
n1/dfs-bfs.py"
Inorder traversal of a binary tree
4 2 1 3 5

Preorder traversal of a binary tree
1 4 2 3 5

Postorder traversal of a binary tree
4 2 3 5 1

BFS traversal of a binary tree
1 2 3 4 5
```