```
# Define a class for the graph nodes
class Node:
    def __init__(self, value):
        self.value = value
        self.neighbors = []

# Perform DFS recursively
def dfs(node, visited):
    visited.add(node)
    print(node.value)

    for neighbor in node.neighbors:
        if neighbor not in visited:
            dfs(neighbor, visited)

# Create nodes
nodeA = Node("A")
nodeB = Node("B")
nodeC = Node("C")
nodeD = Node("D")
nodeE = Node("E")

# Define node neighbors
nodeA.neighbors = [nodeB, nodeC]
nodeB.neighbors = [nodeD, nodeE]
nodeC.neighbors = [nodeB]
nodeD.neighbors = []
nodeE.neighbors = []

# Perform DFS starting from nodeA
visited = set()
dfs(nodeA, visited)
```