```
from collections import deque

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

# Perform BFS
def bfs(start_node):
    visited = set()
    queue = deque([start_node])

    while queue:
        node = queue.popleft()
        visited.add(node)
        print(node.value)

        for neighbor in node.neighbors:
            if neighbor not in visited and neighbor not in queue:
                queue.append(neighbor)

# 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 BFS starting from nodeA
bfs(nodeA)
```