1. Breadth first search:

**Input:**

tree = { 'A':['B', 'C'], 'B':['D', 'E'], 'C':['F', 'G'], 'D':[], 'E':[], 'F':[], 'G':[] }

start = input("Enter the Start node: ")

def bfs(tree, start):

    open\_list = [start]

    close\_list = []

    while open\_list:

        node = open\_list.pop(0)

        if node not in close\_list:

            close\_list.append(node)

            neighbour = tree[node]

            for i in neighbour:

                open\_list.append(i)

    return close\_list

print("Traversal is", bfs(tree, start))

**Output:**



1. Goal Search:

**Input:**

tree = { 'A':['B', 'C'], 'B':['D', 'E'], 'C':['F', 'G'], 'D':[], 'E':[], 'F':[], 'G':[] }

start = input("Enter the Start node: ")

goal = input("Enter Goal State: ")

def bfs(tree, start, goal):

    open\_list = [start]

    close\_list = []

    if start == goal:

        print("Start state is goal state")

        return close\_list

    close\_list.append(start)

    while open\_list:

        node = open\_list.pop(0)

        neighbour = tree[node]

        for i in neighbour:

            if i not in close\_list:

                close\_list.append(i)

                open\_list.append(i)

                if i == goal:

                    return close\_list

    print("Goal not found")

    return []

print("Traversal is:", bfs(tree, start, goal))

**Output:**





