```
#Undirected graph where each node is connected to their adjacent node on both ways
graph = {"francfurt":["mainhain", "wulzburg", "kassel"],
         "mainhain": ["karlsuhe", "francfurt"],
         "wulzburg": ["francfurt", "numburg", "earthfurt"],
         "kassel":   ["francfurt","munchen"],
         "karlsuhe": ["ausburg","mainhain"],
         "numburg":  ["statgaurd", "wulzburg"],
         "ausburg":["karlsuhe"],
         "statgaurd":["numburg"],
         "earthfurt":["wulzburg"],
         "munchen":["kassel"]
        }


visited_list=[]


def DFS2(node1,target_node):
  """
  """
  visited_list.append(node1)
  if node1==target_node:
    return True
  for i in graph[node1]:
    if i not in visited_list:
      if DFS2(i,target_node):
        return True
    return False


DFS2("numburg","kassel")
print(visited_list)
```

```
['numburg', 'statgaurd']
```

```
visited_list=[]


queue_list=[]


def BFS2(node1,target_node):
  """
  """
  visited_list.append(node1)
  if node1==target_node:
    return True
  for i in graph[node1]:
    if i not in visited_list and i not in queue_list:
      queue_list.append(i)
      # if BFS2(i,target_node):
      #   return True
  if queue_list:
    node1=queue_list.pop(0)
    if BFS2(node1,target_node):
      return True
  return False


BFS2("francfurt","munchen")
print(visited_list)
# print(queue_list)
```

```
['francfurt', 'mainhain', 'wulzburg', 'kassel', 'karlsuhe', 'numburg', 'earthfurt', 'munchen']
```