PRACTICAL 1

Aim: To implement BFS Algorithm.

Code:

```
graph = {
    "oradea": [("zerind", 46), ("sibiu", 137)],
    "zerind": [("oradea", 46), ("arad", 43)],
    "sibiu": [("oradea", 137), ("arad", 121), ("rimnicu vikea", 54),
("fagaras", 98)],
    "arad": [("zerind", 43), ("sibiu", 121), ("timisoara", 82)],
    "rimnicu vikea": [("sibiu", 54), ("craiova", 124), ("pitesti", 97)],
    "fagaras": [("sibiu", 98), ("bucharest", 155)],
    "timisoara": [("arad", 82), ("lugoj", 77)],
    "craiova": [("pitesti", 104), ("rimnicu vikea", 124), ("dobreta", 89)],
    "pitesti": [("rimnicu vikea", 97), ("bucharest", 90), ("craiova", 104)],
    "bucharest": [("fagaras", 155), ("pitesti", 90), ("giurgiu", 62),
("urziceni", 61)],
    "lugoj": [("timisoara", 77), ("mehadia", 40)],
    "dobreta": [("craiova", 89), ("mehadia", 40)],
    "giurgiu": [("bucharest", 62)],
    "urziceni": [("bucharest", 61), ("vaslui", 108), ("hirsova", 78)],
    "mehadia": [("lugoj", 40), ("dobreta", 40)],
    "vaslui": [("urziceni", 108), ("lasi", 72)],
    "hirsova": [("urziceni", 78), ("eforic", 64)],
    "lasi": [("vaslui", 72), ("neamt", 74)],
    "eforic": [("hirsova", 64)],
    "neamt": [("lasi", 74)],
}
def bfs_connected_component(graph, start, goal):
    explored = []
    queue = [start]
    cost = [0]
    weight = 0
    if start == goal:
        return "Goal Found"
    while queue:
        node = queue.pop(0)
        a = cost.pop(0)
        if node not in explored:
            explored.append(node)
            weight += a
            neighbours = graph[node]
            for neighbour in neighbours:
                if neighbour[0] not in explored:
                    if neighbour[0] not in queue:
                        queue.append(neighbour[0])
```

```
                        cost.append(neighbour[1])
            print(queue)
            print(cost)
        if node == goal:
            return (explored, weight)
    return "Goal not found"
print(bfs_connected_component(graph, "arad", "bucharest"))
```

Output:

```
[Running] python -u "c:\Users\athar\Documents\Practicals\AI Practical\P1\BFS.py"
['zerind', 'sibiu', 'timisoara']
[43, 121, 82]
['sibiu', 'timisoara', 'oradea']
[121, 82, 46]
['timisoara', 'oradea', 'rimnicu vikea', 'fagaras']
[82, 46, 54, 98]
['oradea', 'rimnicu vikea', 'fagaras', 'lugoj']
[46, 54, 98, 77]
['rimnicu vikea', 'fagaras', 'lugoj']
[54, 98, 77]
['fagaras', 'lugoj', 'craiova', 'pitesti']
[98, 77, 124, 97]
['lugoj', 'craiova', 'pitesti', 'bucharest']
[77, 124, 97, 155]
['craiova', 'pitesti', 'bucharest', 'mehadia']
[124, 97, 155, 40]
['pitesti', 'bucharest', 'mehadia', 'dobreta']
[97, 155, 40, 89]
['bucharest', 'mehadia', 'dobreta']
[155, 40, 89]
['mehadia', 'dobreta', 'giurgiu', 'urziceni']
[40, 89, 62, 61]
(['arad', 'zerind', 'sibiu', 'timisoara', 'oradea', 'rimnicu vikea', 'fagaras', 'lugoj', 'craiova', 'pitesti', 'bucharest'], 897)
```