Artificial Intelligence

Lab 5

Submitted by

Muhammad Bilal

FA21-BCS-110

Comsats University Islamabad,

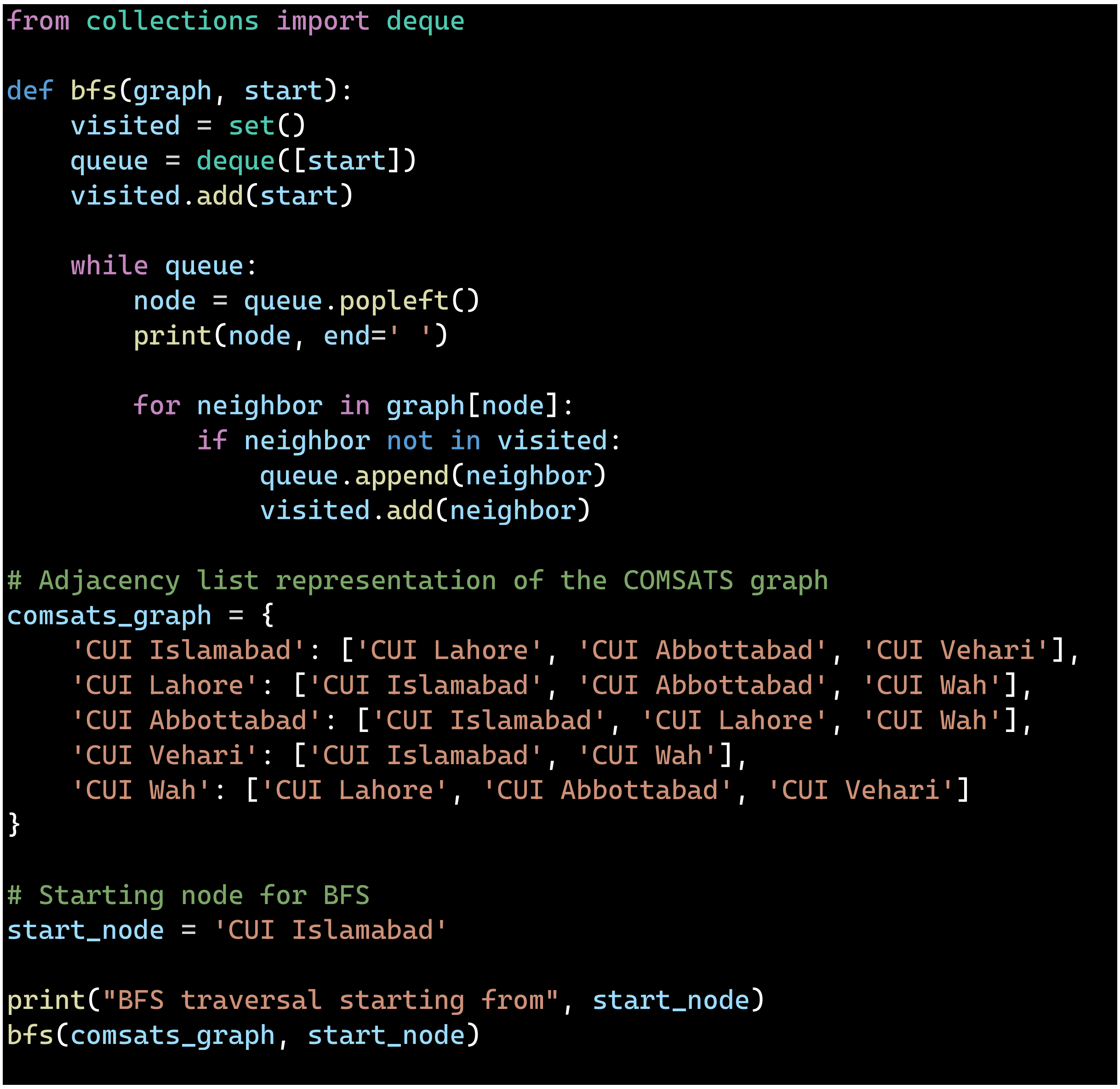
Abbottabad Campus.

Task 1 [10 points]

Implement Breadth-First Search (BFS) for Comsats University. The following is the adjacency list representation of the COMSATS graph.

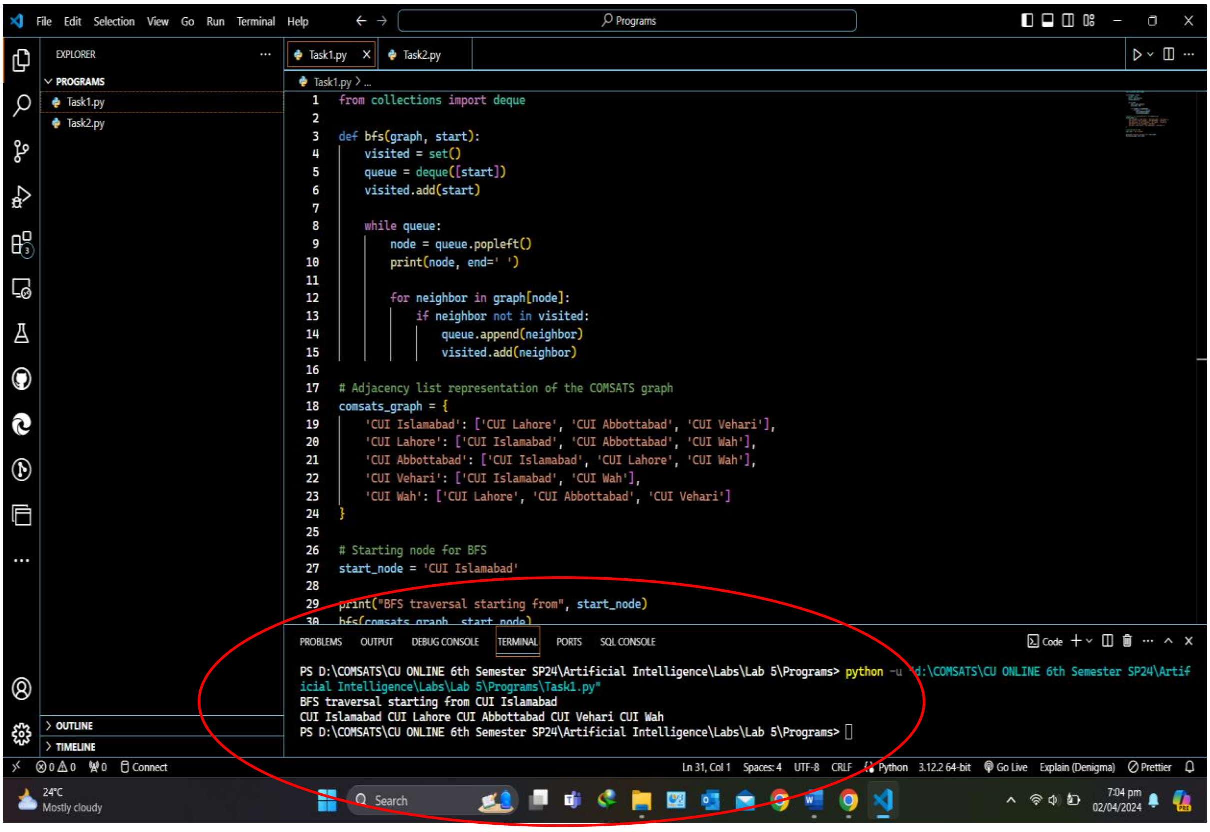
```
comsats_graph = {
    'CUI Islamabad': ['CUI Lahore', 'CUI Abbottabad', 'CUI Vehari'],
    'CUI Lahore': ['CUI Islamabad', 'CUI Abbottabad', 'CUI Wah'],
    'CUI Abbottabad': ['CUI Islamabad', 'CUI Lahore', 'CUI Wah'],
    'CUI Vehari': ['CUI Islamabad', 'CUI Wah'],
    'CUI Wah': ['CUI Lahore', 'CUI Abbottabad', 'CUI Vehari']
}
```

Code:



```python
from collections import deque

def bfs(graph, start):
    visited = set()
    queue = deque([start])
    visited.add(start)

    while queue:
        node = queue.popleft()
        print(node, end=' ')

        for neighbor in graph[node]:
            if neighbor not in visited:
                queue.append(neighbor)
                visited.add(neighbor)

# Adjacency list representation of the COMSATS graph
comsats_graph = {
    'CUI Islamabad': ['CUI Lahore', 'CUI Abbottabad', 'CUI Vehari'],
    'CUI Lahore': ['CUI Islamabad', 'CUI Abbottabad', 'CUI Wah'],
    'CUI Abbottabad': ['CUI Islamabad', 'CUI Lahore', 'CUI Wah'],
    'CUI Vehari': ['CUI Islamabad', 'CUI Wah'],
    'CUI Wah': ['CUI Lahore', 'CUI Abbottabad', 'CUI Vehari']
}

# Starting node for BFS
start_node = 'CUI Islamabad'

print("BFS traversal starting from", start_node)
bfs(comsats_graph, start_node)
```

Output:



File Edit Selection View Go Run Terminal Help
Programs
EXPLORER
PROGRAMS
Task1.py
Task2.py
OUTLINE
TIMELINE
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS SQL CONSOLE
Ln 31, Col 1 Spaces: 4 UTF-8 CRLF Python 3.12.2 64-bit Go Live Explain (Denigma) Prettier
24°C
Mostly cloudy
Search
7:04 pm
02/04/2024

```python
from collections import deque

def bfs(graph, start):
    visited = set()
    queue = deque([start])
    visited.add(start)

    while queue:
        node = queue.popleft()
        print(node, end=' ')

        for neighbor in graph[node]:
            if neighbor not in visited:
                queue.append(neighbor)
                visited.add(neighbor)

# Adjacency list representation of the COMSATS graph
comsats_graph = {
    'CUI Islamabad': ['CUI Lahore', 'CUI Abbottabad', 'CUI Vehari'],
    'CUI Lahore': ['CUI Islamabad', 'CUI Abbottabad', 'CUI Wah'],
    'CUI Abbottabad': ['CUI Islamabad', 'CUI Lahore', 'CUI Wah'],
    'CUI Vehari': ['CUI Islamabad', 'CUI Wah'],
    'CUI Wah': ['CUI Lahore', 'CUI Abbottabad', 'CUI Vehari']
}

# Starting node for BFS
start_node = 'CUI Islamabad'

print("BFS traversal starting from", start_node)
bfs(comsats_graph, start_node)
```

```
PS D:\COMSATS\CU ONLINE 6th Semester SP24\Artificial Intelligence\Labs\Lab 5\Programs> python -u "d:\COMSATS\CU ONLINE 6th Semester SP24\Artif
icial Intelligence\Labs\Lab 5\Programs\Task1.py"
BFS traversal starting from CUI Islamabad
CUI Islamabad CUI Lahore CUI Abbottabad CUI Vehari CUI Wah
PS D:\COMSATS\CU ONLINE 6th Semester SP24\Artificial Intelligence\Labs\Lab 5\Programs> 
```

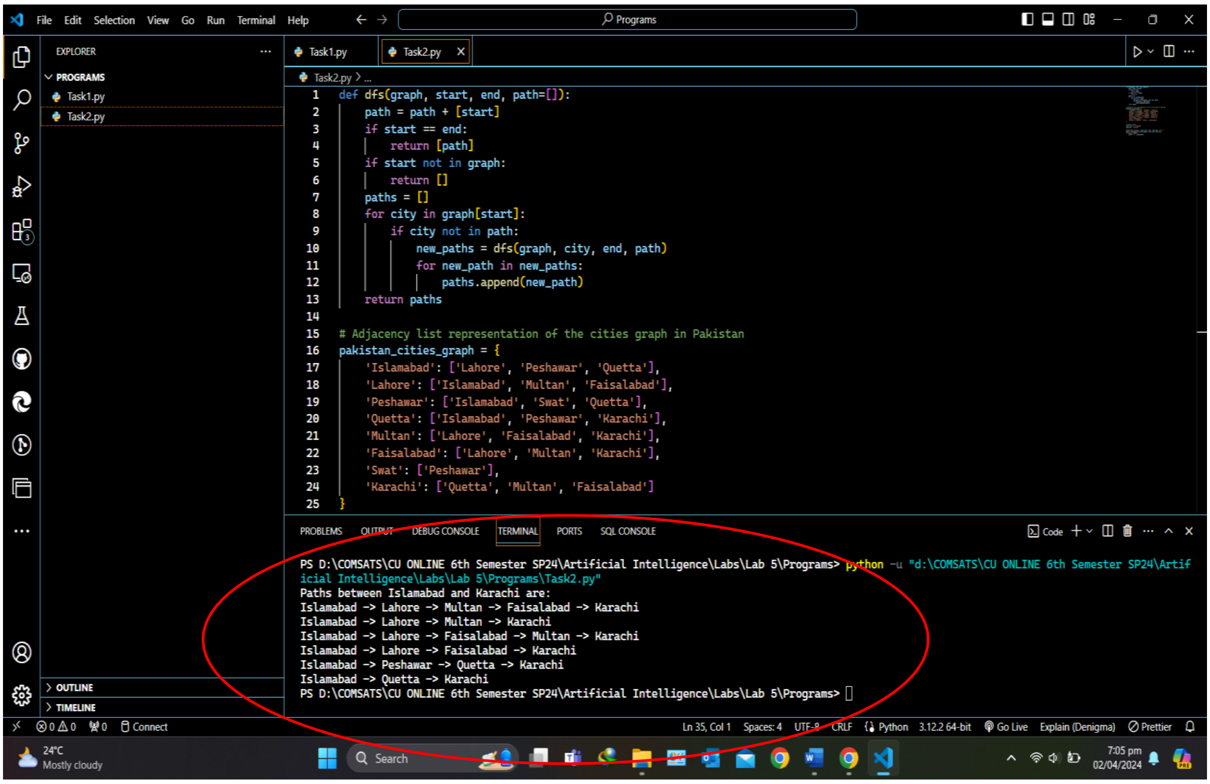
Task 2 [10 points]

Implement Depth-First Search (DFS) to find paths between various cities in Pakistan.

Code:

```python
def dfs(graph, start, end, path=[]):
    path = path + [start]
    if start == end:
        return [path]
    if start not in graph:
        return []
    paths = []
    for city in graph[start]:
        if city not in path:
            new_paths = dfs(graph, city, end, path)
            for new_path in new_paths:
                    paths.append(new_path)
    return paths

# Adjacency list representation of the cities graph in Pakistan
pakistan_cities_graph = {
    'Islamabad': ['Lahore', 'Peshawar', 'Quetta'],
    'Lahore': ['Islamabad', 'Multan', 'Faisalabad'],
    'Peshawar': ['Islamabad', 'Swat', 'Quetta'],
    'Quetta': ['Islamabad', 'Peshawar', 'Karachi'],
    'Multan': ['Lahore', 'Faisalabad', 'Karachi'],
    'Faisalabad': ['Lahore', 'Multan', 'Karachi'],
    'Swat': ['Peshawar'],
    'Karachi': ['Quetta', 'Multan', 'Faisalabad']
}

# Example usage:
start_city = 'Islamabad'
end_city = 'Karachi'

print("Paths between", start_city, "and", end_city, "are:")
paths = dfs(pakistan_cities_graph, start_city, end_city)
for path in paths:
    print(' -> '.join(path))
```

Output:



File Edit Selection View Go Run Terminal Help
Programs
EXPLORER
PROGRAMS
Task1.py
Task2.py
Task2.py > ...
OUTLINE
TIMELINE
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS SQL CONSOLE
Code
Ln 35, Col 1 Spaces: 4 UTF-8 CRLF Python 3.12.2 64-bit Go Live Explain (Denigma) Prettier
24°C Mostly cloudy
Search
7:05 pm 02/04/2024

```
def dfs(graph, start, end, path=[]):
    path = path + [start]
    if start == end:
        return [path]
    if start not in graph:
        return []
    paths = []
    for city in graph[start]:
        if city not in path:
            new_paths = dfs(graph, city, end, path)
            for new_path in new_paths:
                paths.append(new_path)
    return paths

# Adjacency list representation of the cities graph in Pakistan
pakistan_cities_graph = {
    'Islamabad': ['Lahore', 'Peshawar', 'Quetta'],
    'Lahore': ['Islamabad', 'Multan', 'Faisalabad'],
    'Peshawar': ['Islamabad', 'Swat', 'Quetta'],
    'Quetta': ['Islamabad', 'Peshawar', 'Karachi'],
    'Multan': ['Lahore', 'Faisalabad', 'Karachi'],
    'Faisalabad': ['Lahore', 'Multan', 'Karachi'],
    'Swat': ['Peshawar'],
    'Karachi': ['Quetta', 'Multan', 'Faisalabad']
}
```

```
PS D:\COMSATS\CU ONLINE 6th Semester SP24\Artificial Intelligence\Labs\Lab 5\Programs> python -u "d:\COMSATS\CU ONLINE 6th Semester SP24\Artificial Intelligence\Labs\Lab 5\Programs\Task2.py"
Paths between Islamabad and Karachi are:
Islamabad -> Lahore -> Multan -> Faisalabad -> Karachi
Islamabad -> Lahore -> Multan -> Karachi
Islamabad -> Lahore -> Faisalabad -> Multan -> Karachi
Islamabad -> Lahore -> Faisalabad -> Karachi
Islamabad -> Peshawar -> Quetta -> Karachi
Islamabad -> Quetta -> Karachi
PS D:\COMSATS\CU ONLINE 6th Semester SP24\Artificial Intelligence\Labs\Lab 5\Programs> 
```