AA_LAB_09_Assignment

CE_054

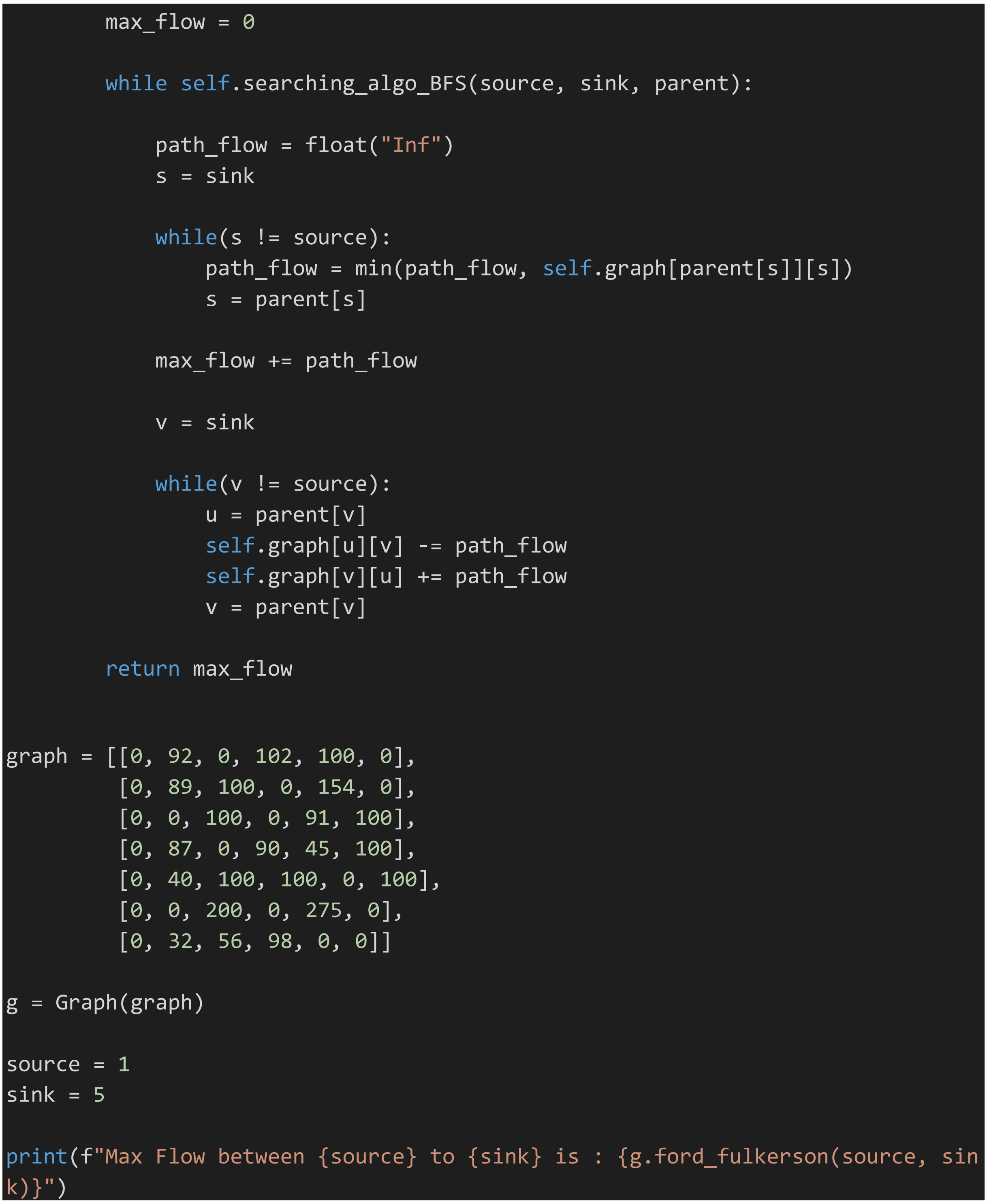
Aim :- Implementation of Algorithm which can gives us maximum flow from network this algorithm called Ford-Fulkerson Algorithm.

Program :-

1. Implementation of Ford Fulkerson Algorithm.

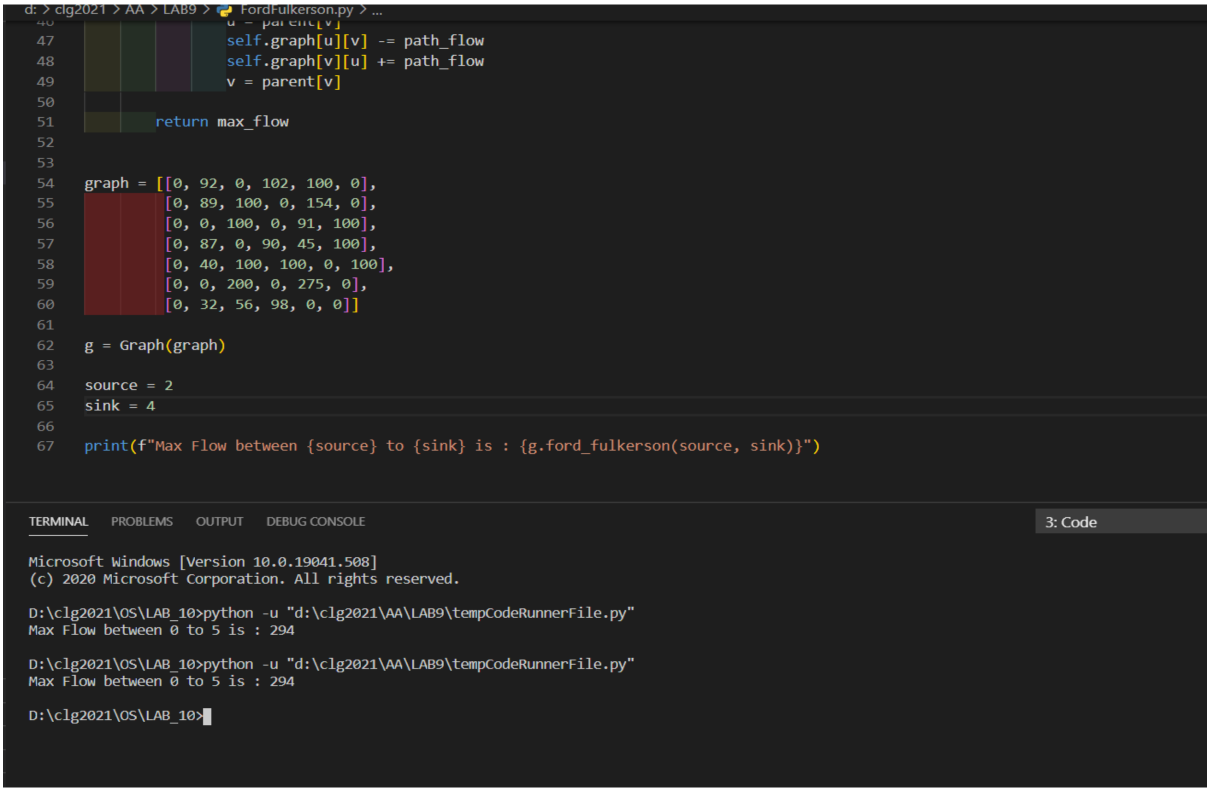
Code :-

```
# -*- coding: utf-8 -*-

"""
Created on Thu Sep 17 21:40:06 2020

@author: DHRUV
"""

from collections import defaultdict

class Graph:

    def __init__(self, graph):
        self.graph = graph
        self.rows = len(graph)

    def searching_algo_BFS(self, s, t, parent):
        visited = [False] * (self.rows)
        queue = []
        queue.append(s)
        visited[s] = True

        while queue:
            u = queue.pop(0)
            for ind, val in enumerate(self.graph[u]):
                if visited[ind] == False and val > 0:
                    queue.append(ind)
                    visited[ind] = True
                    parent[ind] = u

        return True if visited[t] else False

    def ford_fulkerson(self, source, sink):
        parent = [-1] * (self.rows)
```



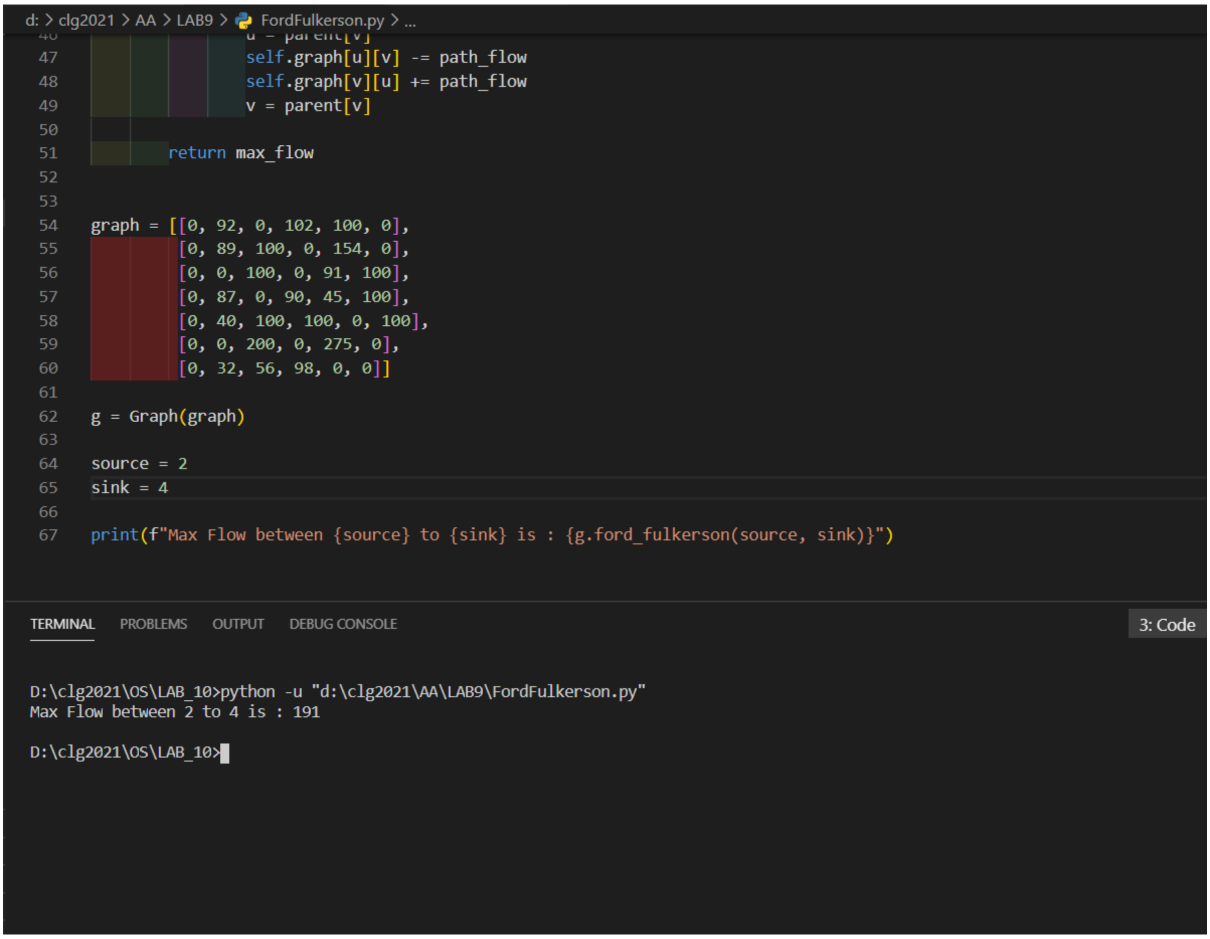
```
        max_flow = 0

        while self.searching_algo_BFS(source, sink, parent):

            path_flow = float("Inf")
            s = sink

            while(s != source):
                path_flow = min(path_flow, self.graph[parent[s]][s])
                s = parent[s]

            max_flow += path_flow

            v = sink

            while(v != source):
                u = parent[v]
                self.graph[u][v] -= path_flow
                self.graph[v][u] += path_flow
                v = parent[v]

        return max_flow


graph = [[0, 92, 0, 102, 100, 0],
         [0, 89, 100, 0, 154, 0],
         [0, 0, 100, 0, 91, 100],
         [0, 87, 0, 90, 45, 100],
         [0, 40, 100, 100, 0, 100],
         [0, 0, 200, 0, 275, 0],
         [0, 32, 56, 98, 0, 0]]

g = Graph(graph)

source = 1
sink = 5

print(f"Max Flow between {source} to {sink} is : {g.ford_fulkerson(source, sin
k)}")
```

Output :-



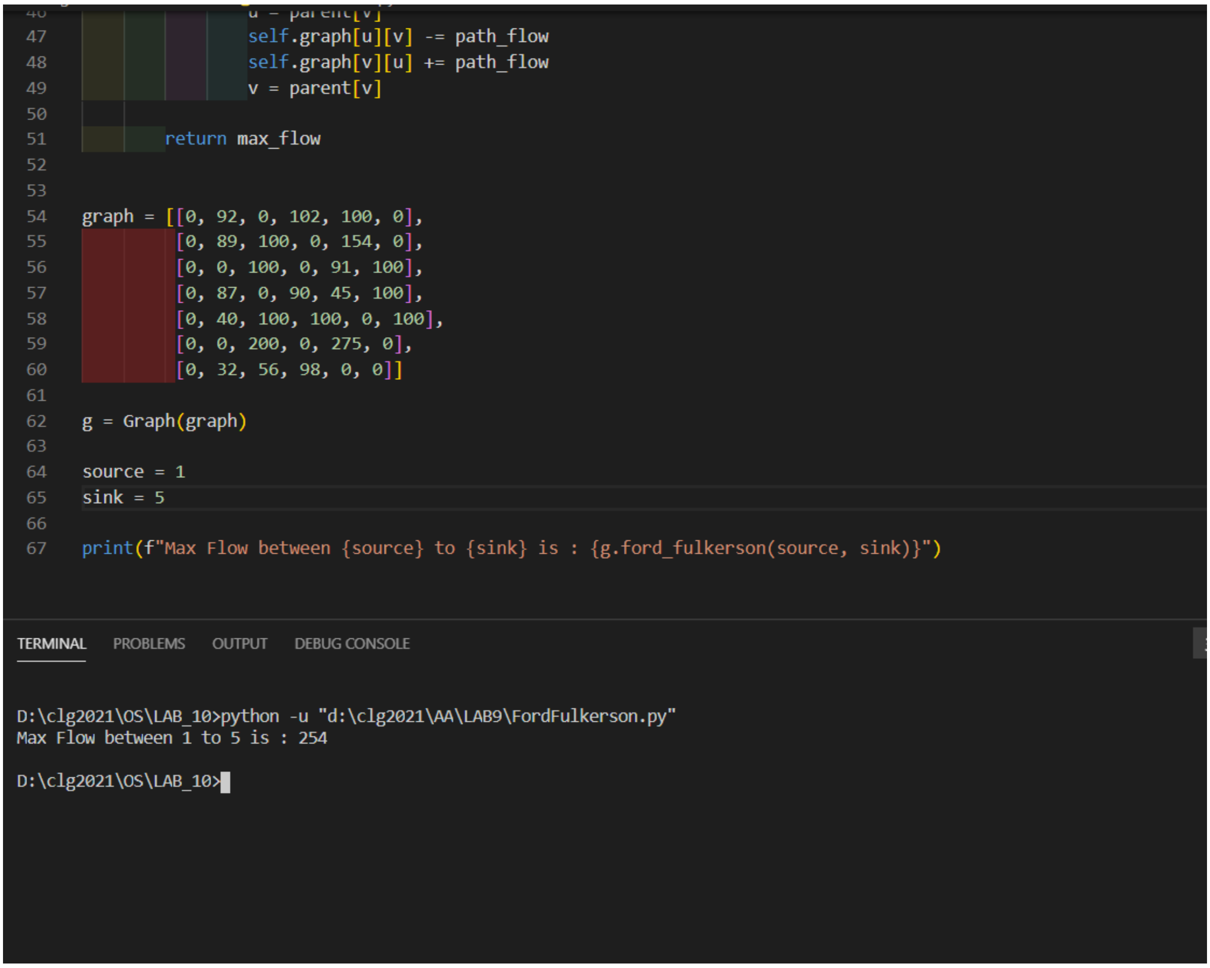
```
                u = parent[v]
                self.graph[u][v] -= path_flow
                self.graph[v][u] += path_flow
                v = parent[v]

        return max_flow


graph = [[0, 92, 0, 102, 100, 0],
         [0, 89, 100, 0, 154, 0],
         [0, 0, 100, 0, 91, 100],
         [0, 87, 0, 90, 45, 100],
         [0, 40, 100, 100, 0, 100],
         [0, 0, 200, 0, 275, 0],
         [0, 32, 56, 98, 0, 0]]

g = Graph(graph)

source = 2
sink = 4

print(f"Max Flow between {source} to {sink} is : {g.ford_fulkerson(source, sink)}")
```

```
Microsoft Windows [Version 10.0.19041.508]
(c) 2020 Microsoft Corporation. All rights reserved.

D:\clg2021\OS\LAB_10>python -u "d:\clg2021\AA\LAB9\tempCodeRunnerFile.py"
Max Flow between 0 to 5 is : 294

D:\clg2021\OS\LAB_10>python -u "d:\clg2021\AA\LAB9\tempCodeRunnerFile.py"
Max Flow between 0 to 5 is : 294

D:\clg2021\OS\LAB_10>
```



```
                u = parent[v]
                self.graph[u][v] -= path_flow
                self.graph[v][u] += path_flow
                v = parent[v]

        return max_flow


graph = [[0, 92, 0, 102, 100, 0],
         [0, 89, 100, 0, 154, 0],
         [0, 0, 100, 0, 91, 100],
         [0, 87, 0, 90, 45, 100],
         [0, 40, 100, 100, 0, 100],
         [0, 0, 200, 0, 275, 0],
         [0, 32, 56, 98, 0, 0]]

g = Graph(graph)

source = 2
sink = 4

print(f"Max Flow between {source} to {sink} is : {g.ford_fulkerson(source, sink)}")
```

```
D:\clg2021\OS\LAB_10>python -u "d:\clg2021\AA\LAB9\FordFulkerson.py"
Max Flow between 2 to 4 is : 191

D:\clg2021\OS\LAB_10>
```



```
                u = parent[v]
                self.graph[u][v] -= path_flow
                self.graph[v][u] += path_flow
                v = parent[v]

        return max_flow


graph = [[0, 92, 0, 102, 100, 0],
         [0, 89, 100, 0, 154, 0],
         [0, 0, 100, 0, 91, 100],
         [0, 87, 0, 90, 45, 100],
         [0, 40, 100, 100, 0, 100],
         [0, 0, 200, 0, 275, 0],
         [0, 32, 56, 98, 0, 0]]

g = Graph(graph)

source = 1
sink = 5

print(f"Max Flow between {source} to {sink} is : {g.ford_fulkerson(source, sink)}")
```

```
D:\clg2021\OS\LAB_10>python -u "d:\clg2021\AA\LAB9\FordFulkerson.py"
Max Flow between 1 to 5 is : 254

D:\clg2021\OS\LAB_10>
```

Theory :-

The Ford-Fulkerson algorithm is an algorithm that tackles the max-flow min-cut problem. That is, given a network with vertices and edges between those vertices that have certain weights, how much "flow" can the network process at a time? Flow can mean anything, but typically it means data through a computer network.

It was discovered in 1956 by Ford and Fulkerson. This algorithm is sometimes referred to as a method because parts of its protocol are not fully specified and can vary from implementation to implementation. An algorithm typically refers to a specific protocol for solving a problem, whereas a method is a more general approach to a problem.

The Ford-Fulkerson algorithm assumes that the input will be a graph, G along with a source vertex, S, and a sink vertex, T. The graph is any representation of a weighted graph where vertices are connected by edges of specified weights. There must also be a source vertex and sink vertex to understand the beginning and end of the flow network.