|  |  |
| --- | --- |
| Name: | Prerna Sunil Jadhav |
| Sap Id: | 60004220127 |
| Class: | T. Y. B. Tech (Computer Engineering) |
| Course: | Advance Algorithm Laboratory |
| Course Code: | DJ19CEL602 |
| Experiment No.: | 06 |

**AIM: Implement Ford Fulkerson Method (Max Flow Network).**

**CODE:**

from collections import defaultdict

class Graph:

    def \_\_init\_\_(self, graph):

        self.graph = graph

        self.ROW = len(graph)

    def bfs(self, s, t, parent):

        visited = [False] \* self.ROW

        queue = []

        queue.append(s)

        visited[s] = True

        while queue:

            u = queue.pop(0)

            for ind, val in enumerate(self.graph[u]):

                if not visited[ind] and val > 0:

                    queue.append(ind)

                    visited[ind] = True

                    parent[ind] = u

        return visited[t], parent

    def ford\_fulkerson(self, source, sink):

        max\_flow = 0

        parent = [-1] \* self.ROW

        while True:

            found\_path, parent = self.bfs(source, sink, parent)

            if not found\_path:

                break

            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

            # Print the augmented path and its minimum value

            path = [sink]

            v = sink

            while v != source:

                u = parent[v]

                path.insert(0, u)

                v = u

            print("Augmented path: ", " -> ".join(str(x) for x in path), " Minimum flow: ", path\_flow)

            v = sink

            while v != source:

                u = parent[v]

                self.graph[u][v] -= path\_flow

                self.graph[v][u] += path\_flow

                v = u

        return max\_flow

graph = [ [0, 2, 3, 0, 0],

        [0, 0, 0, 0, 3],

        [0, 1, 0, 1, 0],

        [0,0,0,0,3],

        [0, 0, 0, 0, 0]]

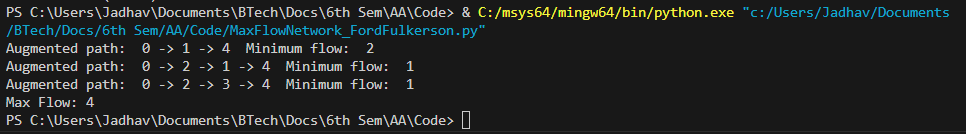
g = Graph(graph)

source = 0

sink = 4

print("Max Flow: %d " % g.ford\_fulkerson(source, sink))

**OUTPUT:**

****