# EXPERIMENT 8

**Objective:** Implementation of Edmonds-Karp algorithm to find maximum flow.

**Brief Theory:**

The Edmonds-Karp algorithm is an implementation of the **Ford-Fulkerson method** for computing the **maximum flow** in a flow network. It uses **Breadth-First Search (BFS)** to find augmenting paths, ensuring the shortest path (in terms of the number of edges) is selected in each iteration.

Key Concepts:

1. Flow Network: A directed graph where each edge has a capacity and a flow.
2. Residual Graph: Tracks the available capacity for each edge after considering the current flow.
3. Augmenting Path: A path in the residual graph from the source to the sink where additional flow can be pushed.

**Algorithm 1** FORDFULKERSON($G$)

```
f ← 0; G_f ← G
while G_f contains an s − t path P do
    Let P be one such path.
    Augment f using P.
    Update G_f
end while
return f
```

**Algorithm 2** EDMONDSKARP($G$)

```
f ← 0; G_f ← G
while G_f contains an s − t path P do
    Let P be an s − t path in G_f with the minimum number of edges.
    Augment f using P.
    Update G_f
end while
return f
```

**Task:**

1) Write a program to perform BFS on a graph and find the shortest path (in terms of the number of edges) from the source to the sink.
2) Implement a function to calculate the bottleneck capacity along a given path and update the flow in the residual graph.
3) Combine BFS and flow augmentation to implement the Edmonds-Karp algorithm and compute the maximum flow in a flow network.
4) Implement Ford-Fulkerson using DFS. Compare it with Edmonds-Karp in terms of execution time.

**Apparatus and components required:** Computer with C or C++ Compiler and Linux/Windows platform.

**Experimental/numerical procedure**: Coding, compilation, editing, run and debugging.