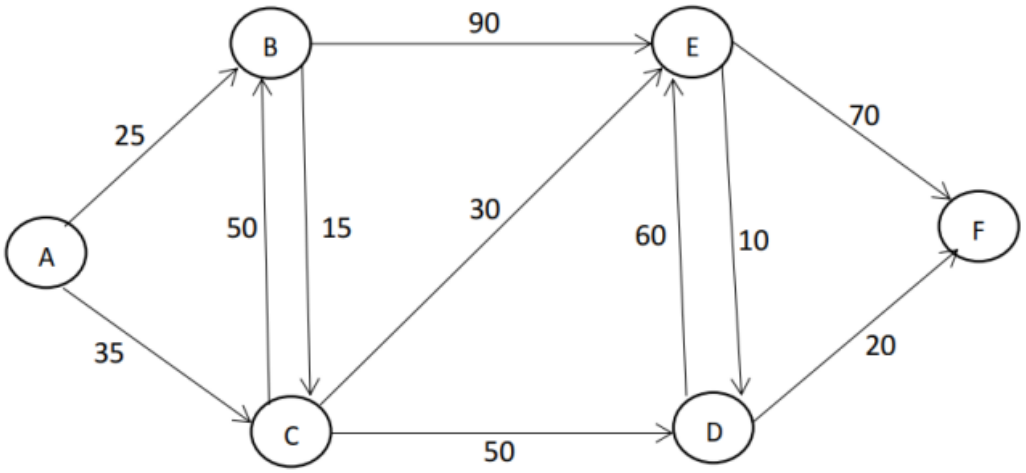
CO323 - Lab 07
Graph Algorithms in Networking: Dijkstra's Algorithm and Bellman-Ford Algorithm

K.G.I.S. Nawarathna
E/17/219

1.



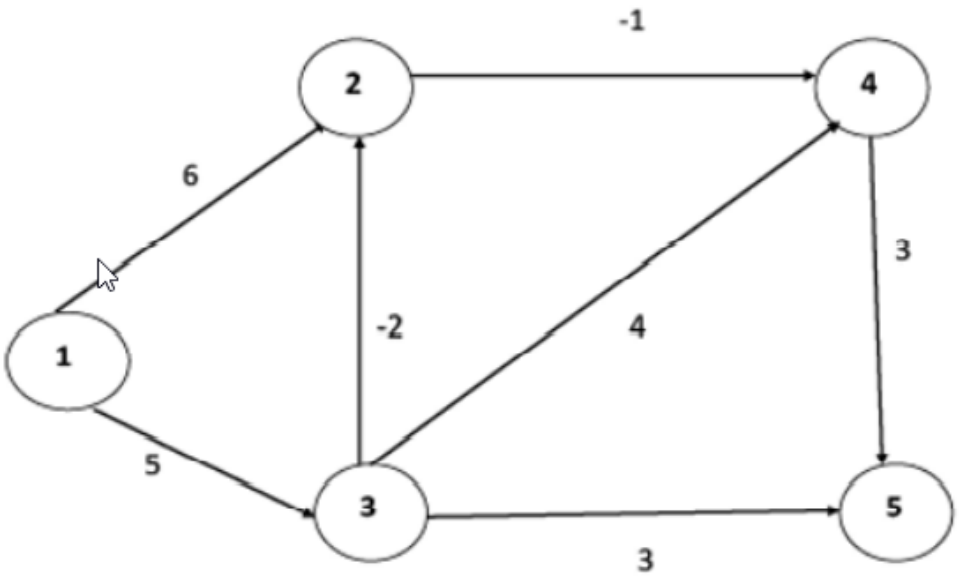
Representation of the graph:

6 6
0 25 35 inf inf inf
inf 0 15 inf 90 inf
inf 50 0 50 30 inf
inf inf inf 0 60 20
inf inf inf 10 0 70
inf inf inf inf inf 0

Pseudocode of the dijkstra's algorithm

```
function Dijkstra(Vertex s) is

      for each Vertex v do
            v.dist := inf
            v.known := false
      s.dist := 0

      while (there is an unknown distance vertex) do

            Vertex v := smallest unknown distance vertex
            v.known := true

            for each Vertex w adjacent to v do

                  if (!w.known) then
                        DistType cvw := cost of edge from v to w

                        if (v.dist + cvw < w.dist) then
                              decrease(w.dist to v.dist + cvw)
                              w.path := v
```

2.



Representation of the graph:

5 5
0 6 5 inf inf
inf 0 inf -1 inf
inf -2 0 4 3
inf inf inf 0 3
inf inf inf inf 0

Pseudo code for the bellman ford algorithm

```
function BellmanFord(list vertices, list edges, vertex source) is


    distance := list of size n
    predecessor := list of size n

    for each vertex v in vertices do

         distance[v] := inf
        predecessor[v] := null

      distance[source] := 0


    repeat |V|-1 times:
         for each edge (u, v) with weight w in edges do
             if distance[u] + w < distance[v] then
                 distance[v] := distance[u] + w
                 predecessor[v] := u

    for each edge (u, v) with weight w in edges do
        if distance[u] + w < distance[v] then
            error "Graph contains a negative-weight cycle"

    return distance, predecessor
```