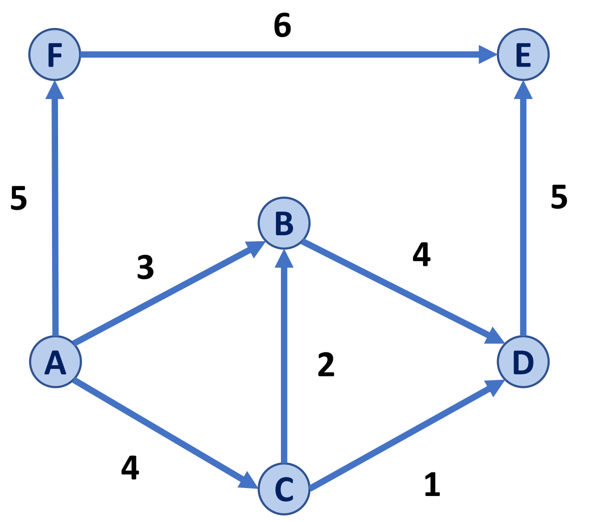
CS 225

Data Structures

April 15 – Floyd-Warshall’s Algorithm

G Carl Evans

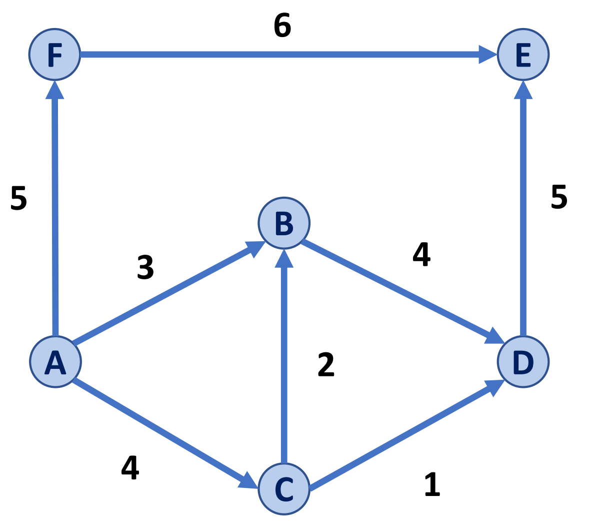
Dijkstra’s Algorithm (SSSP)



	A	B	C	D	E	F
d						
p						

```
Dijkstra(G, s):
  foreach (Vertex v : G):
    d[v] = +inf
    p[v] = NULL
  d[s] = 0

  PriorityQueue Q // min distance, defined by d[v]
  Q.buildHeap(G.vertices())

  repeat n times:
    Vertex u = Q.removeMin()
    T.add(u)
    foreach (Vertex v : neighbors of u not in T):
      if cost(u, v)+ d[u] < d[v]:
        d[v] = cost(u, v) + d[u] //updates PQ
        p[v] = u

```

Dijkstra’s Algorithm (SSSP)



	A	B	C	D	E	F
d						
p						

```
Dijkstra(G, s):
  foreach (Vertex v : G):
    d[v] = +inf
    p[v] = NULL
  d[s] = 0

  PriorityQueue Q // min distance, defined by d[v]
  Q.buildHeap(G.vertices())

  repeat n times:
    Vertex u = Q.removeMin()
    T.add(u)
    foreach (Vertex v : neighbors of u not in T):
      if cost(u, v)+ d[u] < d[v]:
        d[v] = cost(u, v) + d[u] //updates PQ
        p[v] = u

```

Floyd-Warshall Algorithm

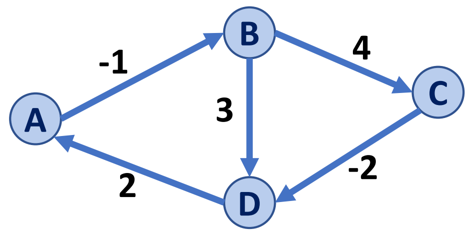
Floyd-Warshall's Algorithm is an alterative to Dijkstra in the presence of negative-weight edges (not negative weight cycles).

```
FloydWarshall(G):
  Let d be a adj. matrix initialized to +inf
  foreach (Vertex v : G):
    d[v][v] = 0
  foreach (Edge (u, v) : G):
    d[u][v] = cost(u, v)

  foreach (Vertex w : G):
    foreach (Vertex u : G):
      foreach (Vertex v : G):
        if (d[u, v] > d[u, w] + d[w, v])
          d[u, v] = d[u, w] + d[w, v]
```

Floyd-Warshall Algorithm

```
FloydWarshall(G):
  Let d be a adj. matrix initialized to +inf
  foreach (Vertex v : G):
    d[v][v] = 0
  foreach (Edge (u, v) : G):
    d[u][v] = cost(u, v)

  foreach (Vertex w : G):
    foreach (Vertex u : G):
      foreach (Vertex v : G):
        if d[u, v] > d[u, w] + d[w, v]:
          d[u, v] = d[u, w] + d[w, v]
```

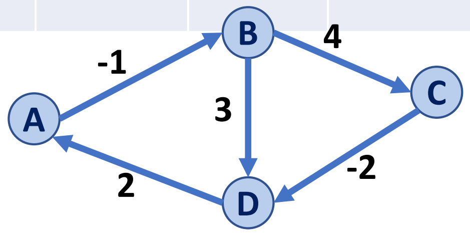
	A	B	C	D
A				
B				
C				
D				



Floyd-Warshall Algorithm

```
foreach (Vertex k : G):
  foreach (Vertex u : G):
    foreach (Vertex v : G):
      if d[u, v] > d[u, k] + d[k, v]:
        d[u, v] = d[u, k] + d[k, v]
```

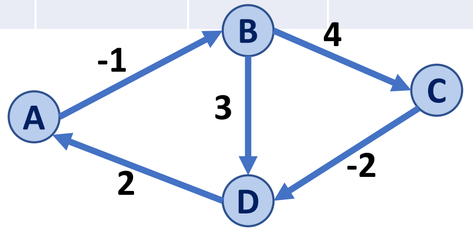
	A	B	C	D
A	0	-1	∞	∞
B	∞	0	4	3
C	∞	∞	0	-2
D	2	∞	∞	0



Floyd-Warshall Algorithm

```
  foreach (Vertex k : G):
    foreach (Vertex u : G):
      foreach (Vertex v : G):
        if d[u, v] > d[u, k] + d[k, v]:
          d[u, v] = d[u, k] + d[k, v]
```

	A	B	C	D
A	0	-1	∞	∞
B	∞	0	4	3
C	∞	∞	0	-2
D	2	∞	∞	0



Let us consider k=A:

$B \rightarrow C$ **4** **vs.** $B \rightarrow A \rightarrow C$ **+∞**

$B \rightarrow D$ **3** **vs.** $B \rightarrow A \rightarrow D$ **+∞**

$C \rightarrow B$ **+∞** **vs.** $C \rightarrow A \rightarrow B$ **+∞**

$C \rightarrow D$ **-2** **vs.** $C \rightarrow A \rightarrow D$ **+∞**

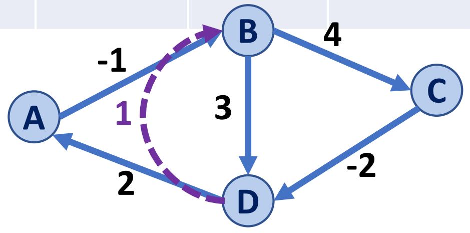
$D \rightarrow B$ **+∞** **vs.** $D \rightarrow A \rightarrow B$

$D \rightarrow C$ **+∞** **vs.** $D \rightarrow A \rightarrow C$

Floyd-Warshall Algorithm

```
foreach (Vertex k : G):
  foreach (Vertex u : G):
    foreach (Vertex v : G):
      if d[u, v] > d[u, k] + d[k, v]:
        d[u, v] = d[u, k] + d[k, v]
```

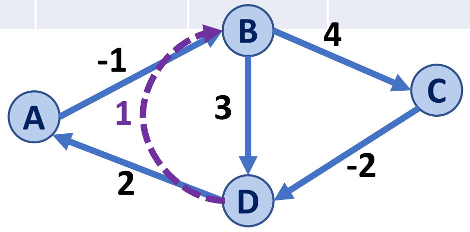
	A	B	C	D
A	0	-1	∞	∞
B	∞	0	4	3
C	∞	∞	0	-2
D	2	1	∞	0



Floyd-Warshall Algorithm

```
foreach (Vertex u : G):
  foreach (Vertex v : G):
    foreach (Vertex k : G):
      if d[u, v] > d[u, k] + d[k, v]:
        d[u, v] = d[u, k] + d[k, v]
```

	A	B	C	D
A	0	-1	∞	∞
B	∞	0	4	3
C	∞	∞	0	-2
D	2	1	∞	0



Let us consider k=B:

$A \rightarrow C$ vs. $A \rightarrow B \rightarrow C$

$A \rightarrow D$ vs. $A \rightarrow B \rightarrow D$

$C \rightarrow A$ vs. $C \rightarrow B \rightarrow A$

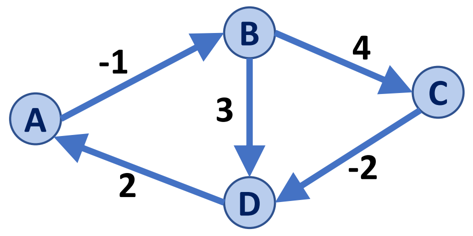
$C \rightarrow D$ vs. $C \rightarrow B \rightarrow D$

$D \rightarrow A$ vs. $D \rightarrow B \rightarrow A$

$D \rightarrow C$ vs. $D \rightarrow B \rightarrow C$

Floyd-Warshall Algorithm

```
FloydWarshall(G):
  Let d be a adj. matrix initialized to +inf
  foreach (Vertex v : G):
    d[v][v] = 0
  foreach (Edge (u, v) : G):
    d[u][v] = cost(u, v)

  foreach (Vertex w : G):
    foreach (Vertex u : G):
      foreach (Vertex v : G):
        if d[u, v] > d[u, w] + d[w, v]:
          d[u, v] = d[u, w] + d[w, v]
```



	A	B	C	D
A				
B				
C				
D				

	A	B	C	D
A				
B				
C				
D				

Floyd-Warshall Algorithm

Running Time?

```
FloydWarshall(G):
  Let d be a adj. matrix initialized to +inf
  foreach (Vertex v : G):
    d[v][v] = 0
  foreach (Edge (u, v) : G):
    d[u][v] = cost(u, v)

  foreach (Vertex u : G):
    foreach (Vertex v : G):
      foreach (Vertex w : G):
        if d[u, v] > d[u, w] + d[w, v]:
          d[u, v] = d[u, w] + d[w, v]
```