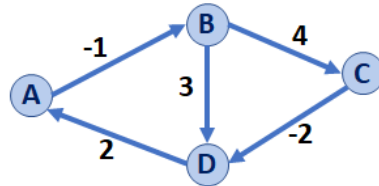


### Floyd-Warshall Algorithm

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

#### Algorithm Design:

- **Goal:** Find the shortest path from vertex **u** to **v**.
- **Setup:** Create an  $n \times n$  matrix that maintains the best known path between every pair of vertices:
  - Initialize  $(u, u)$  to 0.
  - Initialize all edges present on the graph to their edge weight.
  - Initialize all other edges to +infinity.



	A	B	C	D
A				
B				
C				
D				

- For every vertex **k**, consider which of the following are shorter:
  - $\text{path}(u, v)$  - or -
  - $\text{path}(u, k) + \text{path}(k, v)$

#### Big Idea: \_\_\_\_\_

- Store intermediate results to improve build towards an optimal solution.
- Example application of memoization.

#### Running Time:

#### Pseudocode for Floyd-Warshall's Algorithm

```

1 FloydWarshall(G):
2   Input: G, Graph;
3   Output: d, an adjacency matrix of distances between
4   all
5       vertex pairs
6
7   Let d be an adj. matrix (2d array) initialized to +inf
8   foreach (Vertex v : G):
9       d[v][v] = 0
10  foreach (Edge (u, v) : G):
11      d[u][v] = cost(u, v)
12
13  foreach (Vertex u : G):
14      foreach (Vertex v : G):
15          foreach (Vertex w : G):
16              if d[u, v] > d[u, w] + d[w, v]:
17                  d[u, v] = d[u, w] + d[w, v]
18
19  return d
    
```

#### Overview of Graphs:

##### Implementations

- Edge List
- Adjacency Matrix
- Adjacency List

##### Traversals

- Breadth First
- Depth First

##### Minimum Spanning Tree

- Kruskal's Algorithm
- Prim's Algorithm

##### Shortest Path

- Dijkstra's Algorithm
- Floyd-Warshall's Algorithm

...and this is just the beginning. The journey continues to CS 374!

#### CS 225 – Things To Be Doing:

1. MP7 due tonight (April 30); *standard grace period applies.*
2. Final Exam starts Thursday, May 3