Given: Weighted graph $G = (V, E)$ with edge weights $e(u, v)$. May be directed or undirected. May have negative edges, but no negative cycles.

Find: The shortest paths from every vertex to every vertex.

<table>
<thead>
<tr>
<th>A-B: 5</th>
<th>B-A: 5</th>
<th>C-A: 1</th>
<th>D-A: 3</th>
<th>E-A: 10</th>
</tr>
</thead>
<tbody>
<tr>
<td>A-C: 1</td>
<td>B-C: 4</td>
<td>C-B: 4</td>
<td>D-B: 2</td>
<td>E-B: 5</td>
</tr>
<tr>
<td>A-D: 3</td>
<td>B-D: 2</td>
<td>C-D: 2</td>
<td>D-C: 2</td>
<td>E-C: 9</td>
</tr>
<tr>
<td>A-E: 10</td>
<td>B-E: 5</td>
<td>C-E: 9</td>
<td>D-E: 7</td>
<td>E-D: 7</td>
</tr>
<tr>
<td>A-A: 0</td>
<td>B-B: 0</td>
<td>C-C: 0</td>
<td>D-D: 0</td>
<td>E-E: 0</td>
</tr>
</tbody>
</table>
Since $G$ may have negative weighted edges, Bellman-Ford can find shortest path lengths from one vertex to all others. We choose to number vertices 1 to $|V|$, instead of names.

```python
def Many-Bellman-Ford(G=(V,E), l):
    for i in V:
        for j in V:
            dist(i,j) = oo

    for i in V:
        dist(i,*) = Bellman-Ford(G, l, i)
```

Correct? Yes, given correctness of Bellman-Ford.

Complexity? $|V|^2 + |V| \cdot O(\text{Bellman-Ford}) = |V|^2 + |V| \cdot |V| \cdot |E| = O(|V|^3 \cdot |E|)$

Can we do better? Yes, using dynamic programming to remove repetition.
What is the length of the shortest path from vertex i to vertex j, if we only allow direct connections?

What is the length of the shortest path from vertex i to vertex j, if we only allow direct connections, and intermediate paths through vertex l?
The Problem + Subproblems  All Pairs Shortest Paths  Dynamic Programming

\[ \text{dist}(i,j,0) = \ell(i,j) \]  
length from \( i \) to \( j \) without intermediate vertices

\[ \text{dist}(i,j,1) = \min \left\{ \text{dist}(i,j,0), \text{dist}(i,1,0) + \text{dist}(1,j,0) \right\} \]  
length from \( i \) to \( j \) with option of \( 1 \) as an intermediate vertex.

What is next largest sub problem?
subproblems:

\[ \text{dist}(i, j, k) = \text{length of shortest path from } i \text{ to } j, \text{ allowing vertices } 1-k \text{ as possible intermediates.} \]

Recurrence:

\[ \text{dist}(i, j, k) = \min \{ \text{dist}(i, j, k-1) + \text{dist}(j, k, k-1), \text{dist}(i, k, k-1) \} \]

The Problem:

\[ \text{dist}(i, j, n) \]
def Floyd_Warshall(G=(V,E), l):
    for i in V:
        for j in V:
            dist(i, j, 0) = ∞
    for (i, j) in E:
        dist(i, j, 0) = l(i, j)
    for i in V:
        dist(i, i, 0) = 0
    for k=1 to |V|:
        for i in V:
            for j in V:
                dist(i, j, k) = min (dist(i, j, k-1),
                                    dist(i, k, k-1) + dist(k, j, k-1))
$|V|^2 + |E| + |V| + |V| \cdot |V| \cdot |V| \rightarrow O(|V|^3)$