Algorithm: All Shortest Paths (Floyd-Warshall)

```
input : undirected graph G = (V = {1, ..., n}, E)
output: all shortest paths of G

let dist be a |V| × |V| array of minimum distances
        initialized to ∞ (infinity)

for each edge (u, v) ∈ E:
    dist[u][v] ← 1

for each node v:
    dist[v][v] ← 0

for u from 1 to |V|:
    for v from 1 to |V|:
        for w from 1 to |V|:
            if dist[v][w] > dist[v][u] + dist[u][w]:
                dist[v][w] ← dist[v][u] + dist[u][w]
return dist
```

Remark: For $u, v \in V$ $dist[u][v]$ contains the shortest path length from u to v.