Input:
Description of an n-vertex graph via an nxn array w, such that w[u,v] contains weight of edge (u,v). (Weight is infinity if no edge and 0 for w[v,v].) Starting vertex s

Output:
Array A containing...

Goals:
• Design a dynamic programming algorithm for shortest path

1. \( P_{v,i} = \begin{cases} P_{u,i-1} + (u,v) & \text{if shortest path with } i \text{ edges goes through } u \text{ immediately prior to } v \\ P_{w_i-1} + (w,v) & \quad u \quad w \\ \vdots & \quad \vdots \\ P_{v,i-1} & \text{if shortest } s \rightarrow v \text{ path with at most } i \text{ edges uses fewer than } i \text{ edges} \end{cases} \)

\( P_{s,0} = \emptyset \quad P_{v,0} = \text{None} \quad v \in V, \text{ and } v \neq s \)

2. Turn into objective function recurrence relation

\[
L(P_{v,i}) = \text{length of shortest path from } s \rightarrow v, \text{ with at most } i \text{ edges} \\
L(P_{v,i}) = \min_{u \in V} \left( \min_{(u,v) \in E} L(P_{u,i-1}) + w(u,v) \right) \quad L(P_{v,0}) = \infty \\
L(P_{s,0}) = 0 \quad v \neq s \]

3. Write Pseudocode to fill in A (don’t need to work backward)

\[
A[w,2] = L(P_{w,v}) \\
\]

Input: Description of an n-vertex graph via an nxn array w, such that w[u,v] contains weight of edge (u,v). (Weight is infinity if no edge and 0 for w[v,v].) Starting vertex s

Output: Array A containing...?