Goals:

- Design a dynamic programming algorithm for shortest path

Group work

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

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

2. Turn into objective function recurrence relation

$L(P_{v,i})$ = length of shortest path from $s \to v$, with at most i edges

$$L(P_{v,i}) = \min \left\{ \min_{\substack{u \in V \\ u \neq v \\ (u,v) \in E}} \left\{ L(P_{u,i-1}) + w(u,v) \right\}, L_{v,i-1} \right\}$$

$L(P_{s,0}) = 0$ $\quad L(P_{v,0}) = \infty$ $\quad v \neq s$

3. Write Pseudocode to Fill in A (don't need to work backward)

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…?