Last Time on All pair Shortest Path

- Opt substructure. Shortest paths contains shortest subpaths.
- Subprobs are defined by how many edges along a path
- get the predicessor

"Slow shortest path for all pairs"

$$l_{ij}^{(m)} = \min \begin{cases} l_{ij}^{(m-1)} \\ \min\limits_{1 \le k \le |V|} \{ l_{ik}^{(m-1)} \oplus w_{k,j} \} \end{cases}$$

add elemwise

PRED $\pi^{m-1}(i,j)$

$k = \pi^{(m)}(i,j)$

EXTEND.

$$L^{2m}(i,j) = \begin{cases} L^{(m)}(i,j) \\ \min \{ L^{(m)}(i,k) + L^{(m)}(k,j) \} \end{cases}$$

$\pi^{(m)}(i,j)$

TO

$\pi^{(m)}(k,j)$

Floyd-Wharshall Algorithm

Subproblem defined by which verts are allowed along a path (verts are numbered).

$D^{(0)}$:= shortest path with no verts along a path

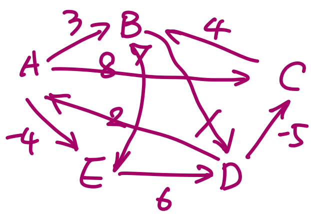
$$= \begin{cases} 0 & i=j \\ w_{ij} & \exists \text{edge } (i,j) \\ \infty & \text{o.w.} \end{cases} \quad \text{and } \pi \quad \begin{matrix} \text{NIL} \\ i \\ \text{NIL} \end{matrix}$$

$D^{(k)}(i,j)$:= shortest path with possibly vertex $1..k$ along the path.

$$= \min \begin{cases} D^{(k-1)}(i,j) & \text{don't help us} \\ D^{(k-1)}(i,k) + D^{(k-1)}(k,j) & \text{we use vertex } k \end{cases}$$

Saving on iteration

Example.



$D^{(0)}$

W	A	B	C	D	E
A	0	3_A	8_A	∞	4_A
B	∞	0	∞	1_B	7_B
C	∞	4_C	0	∞	∞
D	2_D	∞	-5_D	0	∞
E	∞	∞	∞	6_E	0

$D^{(1)}$ — allow go through A

W	A	B	C	D	E
A	0	$\boxed{3_A}$	8_A	∞	4_A
B	∞	0	∞	$\boxed{1_B}$	$\boxed{7_B}$
C	∞	$\boxed{4_C}$	0	∞	∞
D	2_D	$\boxed{5_C}$	-5_D	0	$\boxed{-2_A}$
E	∞	∞	∞	6_E	0

$D^{(2)}$ — allow go through A, B

W	A	B	C	D	E
A	0	3_A	8_A	4_B	4_A
B	∞	0	∞	1_B	7_B
C	∞	4_C	0	$\cancel{\infty}\ 5_B$	$\cancel{\infty}\ 11_B$
D	2_D	5_A	-5_D	0	$\boxed{-2_A}$
E	∞	∞	∞	6_E	0

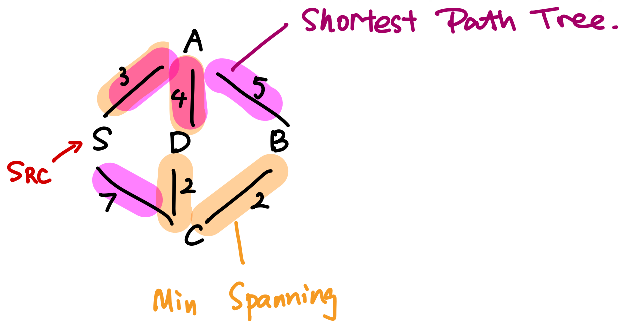
...

finally we will update it to |V|.

- We cant change $L^{(2)}$ to $L^{(4)}$, $L^{(4)}$ to $L^{(8)}$.
- If negative on main diagnal, neg cycle
  no going to stop early.

Prob.

7. Give an example of a weighted undirected graph G (of at least 4 vertices, and with no negative weight cycles) and a starting vertex s in G such that the minimum spanning tree of G is not the same as the shortest path tree of G (starting from s). Remember, the edges used in any single source shortest path solution form a shortest path tree. Show both the minimum spanning tree of G and the shortest path tree of G. Explain which method you used to find the minimum spanning tree and which method you used to find the single source shortest path solution.



6. Your roommate has written a program to implement Dijkstra's shortest path algorithm. Design and analyze a linear time algorithm to check your roommate's algorithm's results. That is, given a graph G = (V,E), a source vertex s, and your roommate's values of v.d (shortest path weight from source to vertex v) and v.pi (predecessor to vertex v for shortest path from source to vertex v) for every vertex v ∈ V, your algorithm must verify their correctness or find a value that is wrong.

- zeros on main diagnal.
- Bellman-Ford, if relax one more time (x).