### In lieu of recitations – 320 Office Hours

<table>
<thead>
<tr>
<th>Mon</th>
<th>Tue</th>
<th>Wed</th>
<th>Thurs</th>
<th>Fri</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td>Cole</td>
</tr>
<tr>
<td>9</td>
<td>Dr. Georg</td>
<td></td>
<td>Dr. Georg</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td></td>
<td></td>
<td>Dr. Georg/ Jim</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>Ali</td>
<td></td>
<td>Jim</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>Cole</td>
<td></td>
<td>Ali</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Cole/ Shannon</td>
<td></td>
<td>Ali</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>Shannon</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Dr. Georg</td>
<td></td>
<td>Dr. Georg</td>
<td>Jim</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Upcoming -- Check Progress page, Piazza postings for updates**

- MST programming assignment due Sat Nov 11 at 11:59pm
  
https://www.youtube.com/watch?v=u0knUHjZbWU&feature=youtu.be
Micro-survey

Why do we add a dummy node to a constraint graph?
• So that we are guaranteed to have a single node from which all other nodes can be reached.

When can we break out of the first Bellman-Ford loop early?
• Whenever nothing changes during a round of relaxation

Why did we keep track of the order of edges we considered in the Bellman-Ford example?
• No reason – I was just experimenting to see if order matters (no), and forgot a few edges in an example. Keeping track of order gave me a counting mechanism so I didn’t forget any edges.
All-Pairs Shortest Paths

CS 320, Fall 2017

Dr. Geri Georg, Instructor
georg@colostate.edu
Applications

Transitive closure of directed graphs
Finding a regular expression denoting the regular language accepted by a finite automaton
Testing whether an undirected graph is bipartite
Flow problems/optimal routing: minimax, maximin
Diameter of a network
Study Aids

• [http://home.cse.ust.hk/faculty/golin/COMP271Sp03/Notes/MyL15.pdf](http://home.cse.ust.hk/faculty/golin/COMP271Sp03/Notes/MyL15.pdf) Mordecai Golin, Hong Kong University of Science and Technology, notes from 2003

• [https://www.youtube.com/watch?v=4OQeCuLYj4](https://www.youtube.com/watch?v=4OQeCuLYj4)
Preliminaries

What does all-pairs shortest paths actually mean?

Can we find all-pairs shortest paths for graphs that have negative cycles?

How about negative weight edges?

How could we figure these out with the algorithms we’ve already studied?
About the Graph

Assume that $G = (V, E)$ is a directed graph. Assume that all vertices are numbered:

$1, 2, \ldots, |V|$, where $n = |V|$

Describe the graph as an $n \times n$ adjacency matrix $W$: $w_{ij} = 0$ if $i = j$, the weight of edge $(i,j)$ if $i \neq j$ and $(i,j) \in E$, and $\infty$ if $(i,j) \notin E$.

Weights can be negative, but there cannot be any negative-weight cycles.
Floyd-Warshall Algorithm

Output matrix $D = d_{ij}$, where $d_{ij}$ contains the weight of a shortest path from $i$ to $j$

Predecessor matrix $\Pi = \pi_{ij}$ where $\pi_{ij}$ is NIL if $i = j$ or there is no path from $i$ to $j$, and otherwise $\pi_{ij}$ is the predecessor of $j$ on some shortest path from $i$

The $i^{th}$ row of $\Pi$ is the shortest-paths tree with root $i$
Observations - 1

If a shortest path $p$ from $u$ to $v$ is this:

$$u \rightarrow x \rightarrow y \rightarrow v$$

Then the portion of $p$ between $x$ and $y$ is a shortest path from $x$ to $y$.

Any subpath of a shortest path is itself a shortest path.
Observations - 2

\{1,2,\ldots,k\} \in V, \text{ for any } i,j \in V \text{ consider all paths from } i \text{ to } j \text{ with intermediate vertices drawn from } \{1,2,\ldots,k\} \text{ and let } p \text{ be a minimum-weight path among them}

Now consider } p \text{ and shortest paths from } i \text{ to } j \text{ with all intermediate vertices in the set } \{1,2,\ldots,k-1\}. \text{ What about } k? 

- If not an intermediate vertex of } p, \text{ then all intermediate vertices of } p \text{ are in } \{1,2,\ldots,k-1\}.
- If an intermediate vertex of } p, \text{ then}

\[ p = i \rightarrow \cdot \rightarrow k \rightarrow \cdot \rightarrow j \] (call these } p_1 \text{ and } p_2 \text{)

and all intermediate vertices of } p_1 \text{ and } p_2 \text{ are in } \{1,2,\ldots,k-1\} \text{ so } p_1 \text{ is a shortest path from } i \text{ to } k \text{ with intermediate vertices in } \{1,2,\ldots,k-1\}; \text{ similarly } p_2
Recursive Floyd-Warshall

\[ d_{ij}^{(k)} = \begin{cases} 
  w_{ij} & \text{if } k = 0 \\
  \min (d_{ij}^{(k-1)}, d_{ik}^{(k-1)} + d_{kj}^{(k-1)}) & \text{if } k \geq 1 
\end{cases} \]

Where \( k \) is the upper bound of the set of intermediate nodes, \( \{1, 2, \ldots, k\} \) and the 2 possibilities are when \( k \) is in the intermediate set of nodes on the shortest path, and when it isn’t.
Bottom-Up Floyd-Warshall

\text{Floyd-Warshall}(W)

1 \quad n = W.\text{rows}
2 \quad D^{(0)} = W
3 \quad \text{for } k = 1 \text{ to } n
4 \quad \quad \text{let } D^{(k)} = (d_{ij}^{(k)}) \text{ be a new } n \times n \text{ matrix}
5 \quad \quad \text{for } i = 1 \text{ to } n
6 \quad \quad \quad \text{for } j = 1 \text{ to } n
7 \quad \quad \quad \quad d_{ij}^{(k)} = \min (d_{ij}^{(k-1)}, d_{ik}^{(k-1)} + d_{kj}^{(k-1)})
8 \quad \text{return } D^{(n)}
Getting to $\Pi$

$k$ not an intermediate vertex on $p$

\[
\pi_{ij}^{(k)} = \begin{cases} 
\pi_{ij}^{(k-1)} & \text{if } d_{ij}^{(k-1)} \leq d_{ik}^{(k-1)} + d_{kj}^{(k-1)} \\
\pi_{kj}^{(k-1)} & \text{if } d_{ij}^{(k-1)} > d_{ik}^{(k-1)} + d_{kj}^{(k-1)}
\end{cases}
\]

$k$ is an intermediate vertex on $p$ so pred is pred of the $kj$ part ($p_2$ from previous slides)
Example

Graph:

- Vertices: 1, 2, 3, 4, 5
- Edges and weights:
  - (1, 2): 3
  - (1, 5): -4
  - (2, 4): 1
  - (2, 5): 2
  - (3, 4): 1
  - (4, 5): 6

Matrix $W$:

\[
\begin{array}{ccccc}
  & 1 & 2 & 3 & 4 & 5 \\
1 & 0 & 3 & 8 & \infty & -4 \\
2 & \infty & 0 & \infty & 1 & 7 \\
3 & \infty & 4 & 0 & \infty & \infty \\
4 & 2 & \infty & \infty & 0 & 8 \\
5 & \infty & \infty & \infty & 6 & 0 \\
\end{array}
\]

Matrix $D^{(0)}$:

\[
\begin{array}{ccccc}
  & 1 & 2 & 3 & 4 & 5 \\
1 & 0 & 3 & 8 & \infty & -4 \\
2 & \infty & 0 & \infty & 1 & 7 \\
3 & \infty & 4 & 0 & \infty & \infty \\
4 & 2 & \infty & \infty & 0 & 8 \\
5 & \infty & \infty & \infty & 6 & 0 \\
\end{array}
\]

Path $\pi$:

- $\pi = \{1, 2, 3, 4, 5\}$
Credits

allpaths: https://www.sc2mapster.com/projects/sc2-map-analyzer/images
https://stackoverflow.com/questions/6799172/negative-weights-using-dijkstras-algorithm/6799344#6799344