Shortest Paths in Graphs

Find shortest path cost from A to *.

Try Dijkstra

<table>
<thead>
<tr>
<th></th>
<th>init</th>
<th>A</th>
<th>E</th>
<th>D</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>∞</td>
<td>5/A</td>
<td>5/A</td>
<td>5/A</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>C</td>
<td>∞</td>
<td>∞/nil</td>
<td>∞/nil</td>
<td>6/D</td>
<td>4/B</td>
<td>X</td>
</tr>
<tr>
<td>D</td>
<td>∞</td>
<td>∞/nil</td>
<td>∞/nil</td>
<td>3/E</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>E</td>
<td>∞</td>
<td>∞/nil</td>
<td>2/A</td>
<td></td>
<td></td>
<td>X</td>
</tr>
</tbody>
</table>

Wrong!

if \( \text{dist}[E,v] > \text{dist}[E,u] + \ell(u,v) \):

\[
\text{dist}[E,v] = \text{dist}[E,u] + \ell(u,v)
\]

Bummer Dijkstra failed.

Why?

Negative edge weights.
Shortest Paths

BFS and Dijkstra Similarity

Initialization

some \( Q \)

while ! \( Q \).empty()

\[ u = Q.pop() \]

for \( (u, v) \in E \):

\[ \text{if } \text{dist}[v] \neq \text{dist}[u] \]

\[ \text{if } \text{dist}[v] > \text{dist}[u] + 1 \]

\[ \text{dist}[v] = \text{dist}[u] + 1 \]

manage \( Q \).
Assume shortest path from $s$ to $t$.

$s \rightarrow v_1 \rightarrow v_2 \rightarrow \ldots \rightarrow v_n \rightarrow t$

Longest number of hops? $|V| - 1$

Cost of this path?

$\sum_{i,j \in \text{Path}} \ell(i, j)$

How to update best cost to each vertex? Use update rule.

What if you update an edge $(u,v)$ where dest $\leq u_j$ has not been minimized yet? No harm, just wasted time as long as we come back later.

Assume no negative cycles.
```
def BF(G, u):
    for v ∈ V:
        dist[v] = ∞
        prev[v] = nil
        dist[u] = 0

    for |V| - 1 times:
        for (u, v) ∈ E:
            if dist[v] > dist[u] + l(u, v):
                dist[v] = dist[u] + l(u, v)
                prev[v] = u
```