General Structure of SSSP algorithms

for $v$ in $V$:

- distance estimates
  $d[v] = \infty$

- predecessor pointers
  $\pi[v] = \text{None}$

$d[s] = 0$  

while $d[v] > d[u] + w(u,v)$ for some $u$:

- distance from $s$ to $s$ is zero

- edge weight

- relax

$d[v] = d[u] + w(u,v)$

$\pi[v] = u$
The algorithm may run for an exponential number of steps:

- Relax $e_1$
- Relax $e_2$
- Relax $\{e_u, \ldots, e_m\}$ recursively to convergence
- Relax $e_3$
- Relax $\{e_u, \ldots, e_m\}$ to convergence again

Number of relaxations, $\Theta(1)$ each:

$$T(3) = 3$$
$$T(n) = 3 + 2T(n-2) = 3 + 6 + 4T(n-4) = 3 + 6 + 12 + 8T(n-6) = \Theta(2^{n/2})$$

And it may fail to terminate:

[Diagram showing a sequence of nodes and edges with labels for relaxation order and distance estimates.]
we must order relaxations for efficiency and
we must add negative-cycle detection (or disallow
negative cycles)

Bellman-Ford

for v in V:
    d[v] = ∞
    Π[v] = None

    d[s] = 0
    do n-1 times:

    for every edge (u,v) in E:
        if d[v] > d[u] + w(u,v):
            d[v] = d[u] + w(u,v)
            Π[v] = u

    for every edge (u,v) in E:
        if d[v] > d[u] + w(u,v):
            report a negative cycle and return
report that there are no negative cycles.
Example:

<table>
<thead>
<tr>
<th>Iteration</th>
<th>0</th>
<th>4</th>
<th>2</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>4</td>
<td>-1</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>2</td>
<td>-2</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>-2</td>
<td></td>
</tr>
<tr>
<td>(nothing changes)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Ordering of edges influences convergence! We did not specify an edge ordering.

Running time of Bellman-Ford:

Initialization: $\Theta(V)$

Main loop: $|V|-1$ iterations over $|E|$ edges,

$\Theta(1)$ operations per edge $= \Theta(V|E|)$

Negative cycle detection $\Theta(E)$

Total # operations is $\Theta(V|E|)$
Two Structural Properties

Theorem: subpaths of shortest paths are also shortest paths.

Proof: Let $p = \langle v_0, v_i, \ldots v_i, v_j, \ldots v_n \rangle$

$p_{ij} = \langle v_i, v_{i+1}, \ldots, v_j \rangle$

$p = \langle v_0, \ldots, v_i, v_{i+1}, \ldots, v_j, \ldots, v_n \rangle$

If $w(p_{ij}) < w(p_{ij})$, then $p$ is not shortest, we can replace $p_{ij}$ with $p_{ij}$ and get $p = \langle v_0, \ldots, v_k, \ldots, v_n \rangle$ with $w(p) < w(p)$.

Theorem: triangle inequality, for all $u, v, x \in V$

we have $\delta(u, v) \leq \delta(u, x) + \delta(x, v)$

Proof:

\[ u \quad \delta(u,v) \quad v \]

\[ \delta(u,x) \quad \delta(x,v) \quad x \]
Analysis of Bellman-Ford

Theorem: if $G=(V,E)$ contains no negative cycles, then at the end of the "do $n-1$ times" loop we have $d[v] = \delta(s,v)$.

Proof: Let $p = (s, v_1, v_2, \ldots, v_k)$ be a shortest path from $s$ to $v_k$.

By the subpath theorem, $(s, v_1, \ldots, v_{k-1})$ is also a SP, so $\delta(s,v_k) = \delta(s,v_{k-1}) + w(v_{k-1}, v_k)$, and similarly $\delta(s,v_i) = \delta(s,v_{i-1}) + w(v_{i-1}, v_i)$.

After 2 iterations of the loop, $d[v_2] = \delta(s,v_2)$

$2$

$\vdots$

$k$

$\vdots$

$\vdots$

$\vdots$

Since no negative cycles, $k \leq n-1$. Done.
Theorem: Bellman-Ford correctly reports negative cycles.

Proof: After $n-1$ iterations, $d[v]$ is the length of the shortest path with $n-1$ or fewer edges from $s$ to $v$.

If $(u,v)$ can still be relaxed, there is a shorter path with $n$ edges; it must contain a cycle.

Black path shorter than red path, so the cycle must have negative weight.

On the other hand, if there is a negative cycle, there will always be an edge that can be relaxed.