Dijkstra’s Algorithm
SSSP, non-neg

Edge weights = $w(x,y)$

Final distances = $d(x,y) = d_w(x,y)$
Dijkstra’s Algorithm
SSSP, non-neg

x ← extractmin

// L(x) is the distance of s to x
// mark x as final

"relax" all the edges out of x
L(y) ← min ( L(y), L(x) + w(x,y) )

\[
\begin{array}{c|ccccc}
  & s & b & c & d & e \\
  \hline
  L(x) & 0 & \infty & \infty & \infty & \infty \\
\end{array}
\]
Dijkstra’s Algorithm
SSSP, non-neg

x <- extractmin
// L(x) is the distance of s to x
// mark x as final

"relax" all the edges out of x
L(y) <- min ( L(y), L(x) + w(x,y) )
Dijkstra’s Algorithm
SSSP, non-neg

x <= extractmin

// L(x) is the distance of s to x
// mark x as final

"relax" all the edges out of x
L(y) <= min ( L(y), L(x) + w(x,y) )
Dijkstra’s Algorithm
SSSP, non-neg

x <- extractmin

// L(x) is the distance of s to x
// mark x as final

"relax" all the edges out of x

L(y) <- min ( L(y), L(x) + w(x,y) )
Dijkstra’s Algorithm
SSSP, non-neg

x <- extractmin

// L(x) is the distance of s to x
// mark x as final

"relax" all the edges out of x
L(y) <- min ( L(y), L(x) + w(x,y) )
Dijkstra’s Algorithm
SSSP, non-neg

m decreasekeys
n extractmins

Fib heap: $O(m + n \log n)$

$x \leftarrow \text{extractmin}$

// $L(x)$ is the distance of $s$ to $x$
// mark $x$ as final

"relax" all the edges out of $x$

$L(y) \leftarrow \min ( L(y), L(x) + w(x,y) )$
Bellman-Ford-Moore Algorithm
SSSP, neg wts OK
Bellman-Ford-Moore Algorithm
SSSP, neg wts OK

n rounds
m time per round
O(mn) time

<table>
<thead>
<tr>
<th></th>
<th>s</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
</tr>
</thead>
<tbody>
<tr>
<td>L(x)</td>
<td>0</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
</tr>
</tbody>
</table>

// L(x) is an upper bound on d(s,x)

for t = 1 to n-1
for all vertices x
    // "relax" all the edges out of x
    L(y) <- min ( L(y), L(x) + w(x,y) )

Claim: if graph has non negative cycles, B-F-M is OK.

Proof: induction. At end of round t, L(y) is shortest path using at most t edges.
Bellman-Ford-Moore Algorithm
SSSP, neg wts OK

n rounds
m time per round
O(mn) time

<table>
<thead>
<tr>
<th></th>
<th>s</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
</tr>
</thead>
<tbody>
<tr>
<td>L(x)</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>5</td>
<td>4</td>
</tr>
</tbody>
</table>

// L(x) is an upper bound on d(s,x)

for t = 1 to n-1
    for all vertices x
        // "relax" all the edges out of x
        L(y) <- min ( L(y), L(x) + w(x,y) )

Claim: if graph has non negative cycles, B-F-M is OK.

Proof: induction. At end of round t, L(y) is shortest path using at most t edges.
Bellman-Ford-Moore Algorithm
SSSP, neg wts OK

n rounds
m time per round
O(mn) time

\[
\begin{array}{cccccc}
  & s & b & c & d & e \\
  L(x) & 0 & 2 & 2 & 5 & 4 \\
\end{array}
\]

// L(x) is an upper bound on w(s,x)

for t = 1 to n-1
  for all vertices x
    // "relax" all the edges out of x
    \[ L(y) \leftarrow \min \left( L(y), L(x) + w(x,y) \right) \]

Claim: If at end, some edge xy is “over-tight”
      ( it has L(y) > L(x) + w(x,y) )
then graph has negative cycle.
Lots more work!
E.g., extensions and implementations of (just) Dijkstra’s algorithm:

<table>
<thead>
<tr>
<th>Expression</th>
<th>Author(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$O(m + n^2)$</td>
<td>Dijkstra’59</td>
</tr>
<tr>
<td>$O(m \log n)$</td>
<td>William’64</td>
</tr>
<tr>
<td>$O(m + n \log n)$</td>
<td>Fredman and Tarjan’87</td>
</tr>
<tr>
<td>$O(m \sqrt{\log n})$</td>
<td>Fredman and Willard’93</td>
</tr>
<tr>
<td>$O(m + n \frac{\log n}{\log \log n})$</td>
<td>Fredman and Willard’94</td>
</tr>
<tr>
<td>$O(m \log \log n)$</td>
<td>Thorup’96</td>
</tr>
<tr>
<td>$O(m + n \sqrt{\log n}^{1+\varepsilon})$</td>
<td>Thorup’96</td>
</tr>
<tr>
<td>$O(m + n \frac{3}{\log n}^{1+\varepsilon})$</td>
<td>Raman’97</td>
</tr>
<tr>
<td>$O(m + n \frac{3}{\log n}^{1+\varepsilon})$</td>
<td>Raman’97</td>
</tr>
<tr>
<td>$O(m \sqrt{\log \log n})$</td>
<td>Han and Thorup’02</td>
</tr>
<tr>
<td>$O(m + n \log \log n)$</td>
<td>Thorup’03</td>
</tr>
<tr>
<td>$O(m \log \log C')$</td>
<td>van Emde Boas’77</td>
</tr>
<tr>
<td>$O(m + n \sqrt{\log C'})$</td>
<td>Ahuja et.al.’90</td>
</tr>
<tr>
<td>$O(m + n \frac{3}{\log C} \log \log C)$</td>
<td>Cherkassky et.al.’97</td>
</tr>
<tr>
<td>$O(m + n \frac{4}{\log C}^{1+\varepsilon})$</td>
<td>Raman’97</td>
</tr>
<tr>
<td>$O(m + n \log \log C')$</td>
<td>Thorup’03</td>
</tr>
</tbody>
</table>
All Pairs SP

Non-neg weights:
  n Dijkstras \quad O(mn + n^2 \log n)

Neg weights:
  n B-F-M \quad O(mn^2)
Johnson’s Algorithm
APSP, neg wts OK

Find “feasible potentials” $\Phi(x)$ such that the “reduced weights”
$$\hat{w}(x,y) := \Phi(x) + w(x,y) - \Phi(y) \geq 0$$

Fact: reduced weights $\hat{w}$ non-neg

Claim: $d_w(x,y) = d_{\hat{w}}(x,y) + \Phi(y) - \Phi(x)$

So shortest paths don’t change, though their weights might.
$\Rightarrow$ suffices to find APSP in this non-neg weights graph!

How to find these “feasible potentials”?
Johnson’s Algorithm
APSP, neg wts OK

Find “feasible potentials” $\Phi(x)$ such that the “reduced weights”
$$\hat{w}(x,y) := \Phi(x) + w(x,y) - \Phi(y) \geq 0$$

Fact: reduced weights $\hat{w}$ non-neg

Claim: $d_w(x,y) = d_{\hat{w}}(x,y) + \Phi(y) - \Phi(x)$

So shortest paths don’t change, though their weights might.
$\Rightarrow$ suffices to find APSP in this non-neg weights graph!

How to find these “feasible potentials”? 

Shortest paths from Z.
No negative cycles created, so B-F-M works.
**All Pairs SP**

Non-neg weights:
- $n$ Dijkstras
  - $O(mn + n^2 \log n)$

Neg weights:
- $n$ B-F-M
  - $O(mn^2)$

Neg weights:
- B-F-M + $n$ Dijkstras
  - $O(mn + n^2 \log n)$

Neg weights:
- Floyd-Warshall
  - $O(n^3)$

for all vertices $z$
for all vertices $x,y$

$$d(x,y) \leftarrow \min \{ d(x,y), d(x,z) + d(z,y) \}$$
### All Pairs SP

<table>
<thead>
<tr>
<th>Type</th>
<th>Algorithm</th>
<th>Time Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Non-neg weights:</td>
<td>n Dijkstra's</td>
<td>(O(mn + n^2 \log n))</td>
</tr>
<tr>
<td>Neg weights:</td>
<td>n Bellman-Ford-Moore</td>
<td>(O(mn^2))</td>
</tr>
<tr>
<td>Neg weights:</td>
<td>Bellman-Ford-Moore + n Dijkstra's</td>
<td>(O(mn + n^2 \log n))</td>
</tr>
<tr>
<td>Neg weights:</td>
<td>Floyd-Warshall</td>
<td>(O(n^3))</td>
</tr>
<tr>
<td>Neg weights:</td>
<td>Naïve Min-Sum-Product</td>
<td>(O(n^3 \log n))</td>
</tr>
</tbody>
</table>