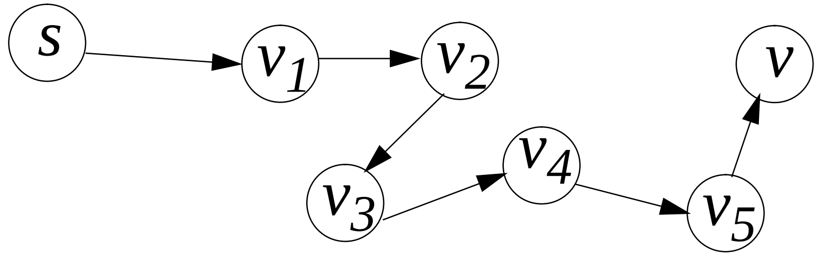
Running time: O(V*E)

Proof of correctness:



Let *v* be reachable from *s*, and let $p = \langle v_0, v_1, \ldots, v_k \rangle$, $v_0 = s$ and $v_k = v$, be a shortest path from *s* to *v*. Since *p* is a simple path, *k*<|V|.

We want to prove by induction for *i=0,1,...,k* that $d[v_i] = \delta(s, v_i)$ after the *i*-th execution of the for loop.

Basis: $d[v_0] = \delta(s, v_0) = 0$

Inductive step: if $d[v_{i-1}] = \delta(s, v_{i-1})$ after *(i-1)*-st pass,

prove that $d[v_i] = \delta(s, v_i)$ after the *i*-th pass

Proof: edge (v_{i-1}, v_i) is relaxed during the *i*-th pass,

and thus by a previous lemma, $d[v_i] = \delta(s, v_i)$

Bellman-Ford(G,w,s)

Initialize-Single-Source(G,s)

for i=1 to |V|-1

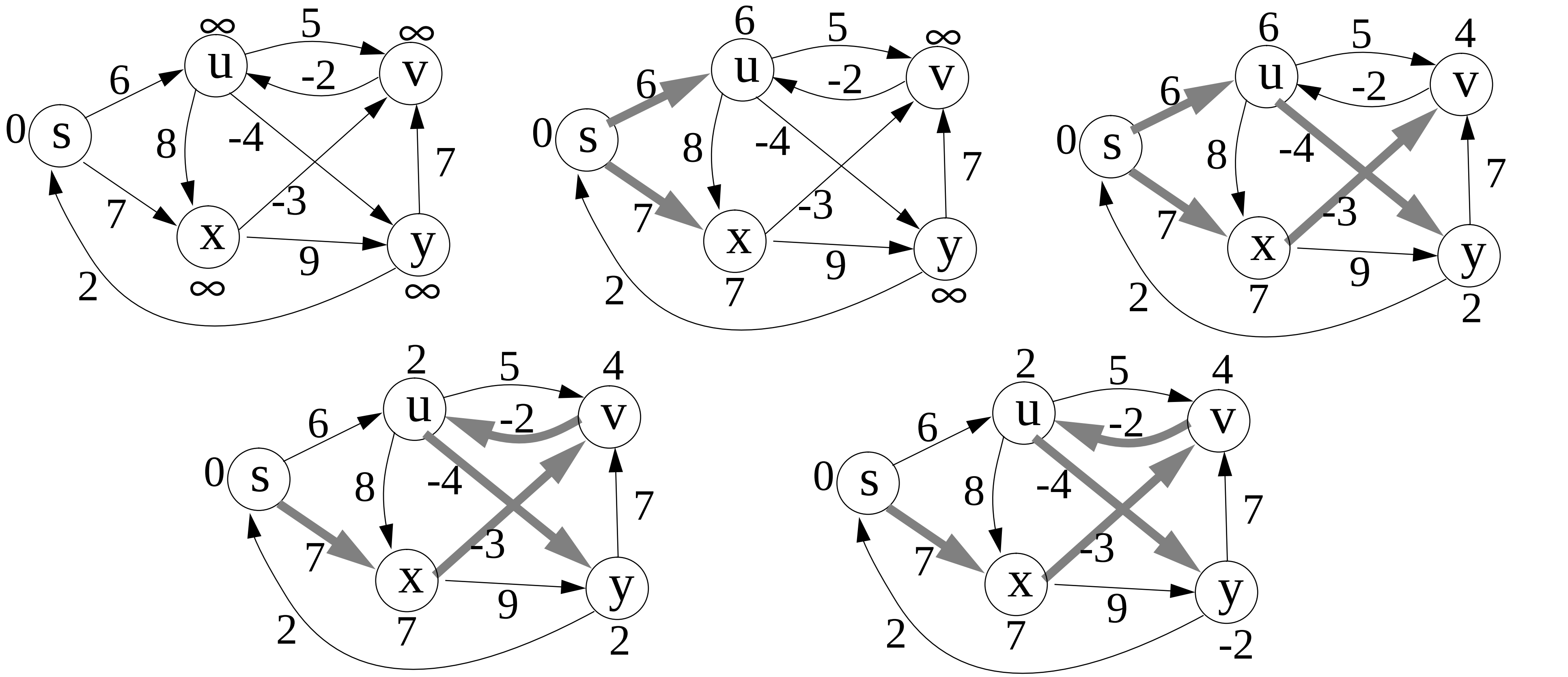
for each edge $(u, v) \in E$ do Relax(u,v,w)

for each edge $(u, v) \in E$

if d[v]>d[u]+w(u,v) then return FALSE (i.e. negative weight cycle)

return TRUE

order of edges: (u,v) (u,x) (u,y) (v,u) (x,v) (x,y) (y,v) (y,z) (z,u) (z,x)



Running time: if we use a table for Q, we get O(V+V*V+E)=O(V^2)

if we use a balanced binary search tree for Q,

we get O(V+V lg V+E lg V)=O(V+E lg V)

Proof of correctness: We want to show that for each vertex $u \in V$, we have $d[u] = \delta(s, u)$ at the time when u is inserted into F. By contradiction.

Let u be the first vertex for which $d[u] \neq \delta(s, u)$ when u is inserted into F.

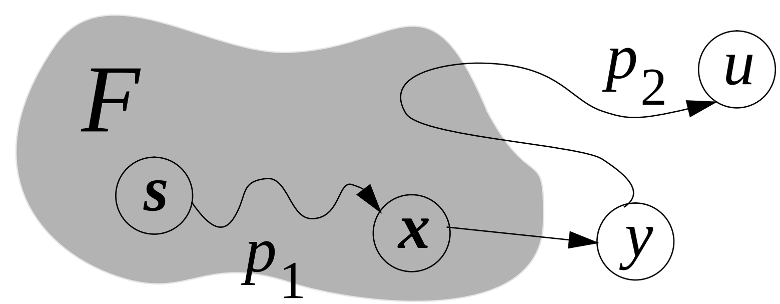
Examine the time <u>at the beginning of the while loop when *u* is inserted into F</u>.

If there is no path from s to u, then $d[u] = \delta(s, u) = \infty$, a contradiction.

So, there is a path from s to u, and let p be the shortest s-to-u path.

p connects a vertex in F, i.e. s, to a vertex in V-F, i.e. u. So, p crosses border.

Look at the first vertex y along p such that $y \in V - F$, and at vertex x preceeding y in p. $x \in F$.



When x was inserted into F, $d[x] = \delta(s, x)$.

Then we did Relax(x,y,w).

By lemma, we have $d[y] = \delta(s, y)$.

$d[y] = \delta(s, y) \leq \delta(s, u) \leq d[u]$

Since both u and y were in V-F when u was chosen, $d[u] \leq d[y]$. Then,

$d[y] = \delta(s, y) = \delta(s, u) = d[u]$, which contradicts our choice of u.

Dijkstra(G,w,s)

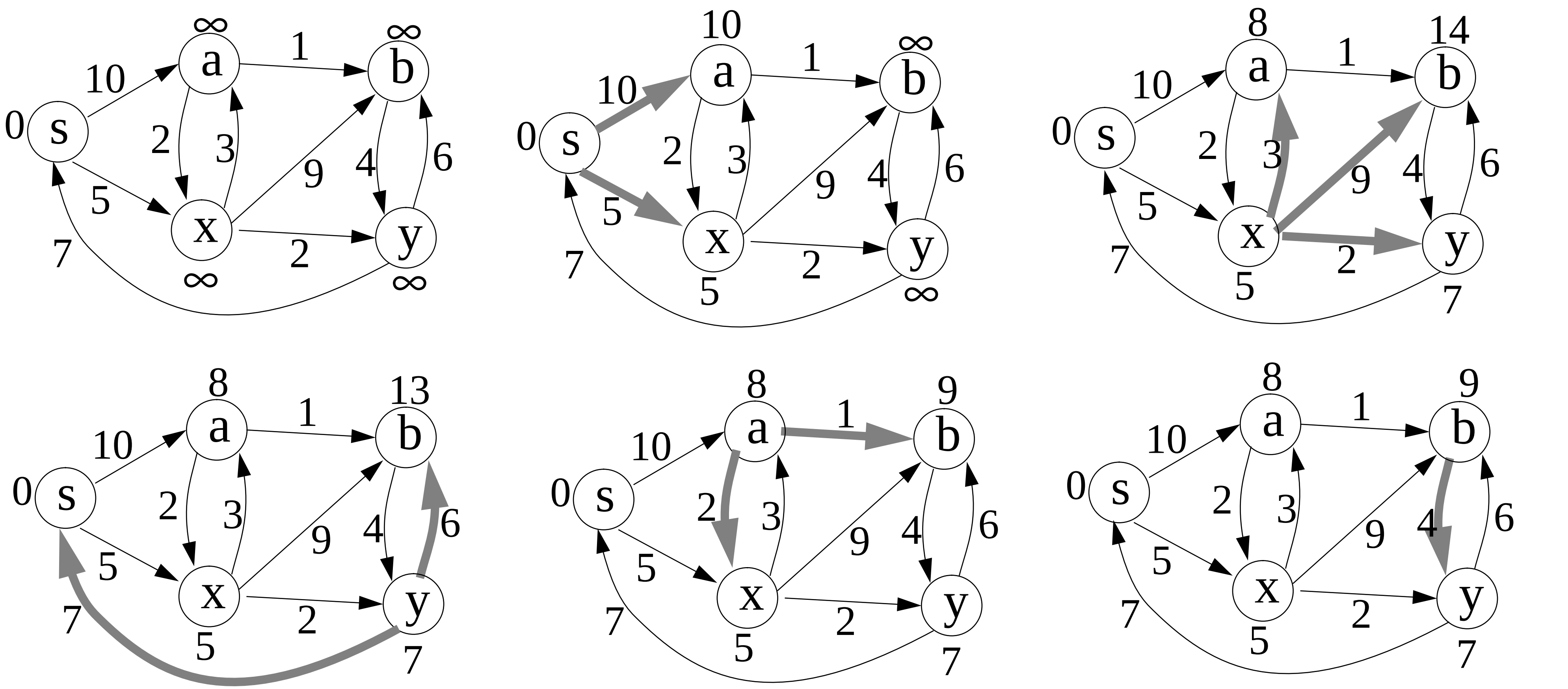
Initialize-Single-Source(G,s); $F=\emptyset$; Q=V

while $Q \neq \emptyset$

u=Extract-Min(Q)

$F = F \cup \{u\}$

for each vertex $v \in Adj[u]$ do Relax(u,v,w)



Lemma: Suppose that the shortest path from *s* to *v* uses edge (*u*,*v*), and G is initialized with Initialize-Single-Source, and then a sequence of relaxations is executed on edges of G that includes Relax(*u*,*v*,*w*).

If $d[u] = \delta(s, u)$ at any time prior to this call Relax(*u*,*v*,*w*),

then $d[v] = \delta(s, v)$ at all times after the call

Proof: By previous Lemma, if $d[u] = \delta(s, u)$ at some point prior to this Relax(*u*,*v*,*w*), then $d[u] = \delta(s, u)$ holds always thereafter.

Thus, after this Relax(*u*,*v*,*w*), we have

$$d[v] \le d[u] + w(u, v) = \delta(s, u) + w(u, v) = \delta(s, v)$$

Thus, $d[v] \le \delta(s, v)$. But from previous Lemma, we have $d[v] \ge \delta(s, v)$.

Thus, $d[v] = \delta(s, v)$

Lemma: Immediately after Relax(u,v,w), we have $d[v] \le d[u] + w(u, v)$

Proof: If before execution of Relax(u,v,w), we had $d[v] > d[u] + w(u, v)$, then $d[v] = d[u] + w(u, v)$ afterwards. If instead, before execution of Relax(u,v,w) we had $d[v] \le d[u] + w(u, v)$, then nothing was changed by Relax(u,v,w) and we still have $d[v] \le d[u] + w(u, v)$.

Lemma: After Initialize-Single-Source, $d[v] \ge \delta(s, v)$ for all vertices v, and this invariant is maintained over any sequence of relaxations on edges of G. And, once d[v] achieves its lower bound of $\delta(s, v)$, it never changes.

Proof: First, $d[v] \ge \delta(s, v)$ is true just after Initialize-Single-Source. We show by contradiction that the invariant holds over any sequence of relaxations. Let v be the first vertex for which a relaxation step of some edge (u,v) causes $d[v] < \delta(s, v)$. Then after this Relax(u,v,w), we have

$d[u] + w(u, v) = d[v] < \delta(s, v) \le \delta(s, u) + w(u, v)$,

which implies that $d[u] < \delta(s, u)$.

This contradicts our assumption that v is the first to violate the invariant. Thus, $d[v] \ge \delta(s, v)$ for all vertices v.

Initialize-Single-Source(G,s)

for each vertex v in V

$d[v] = \infty$

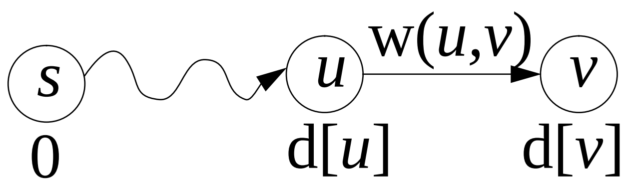
$\pi[v] = NIL$

d[s]=0

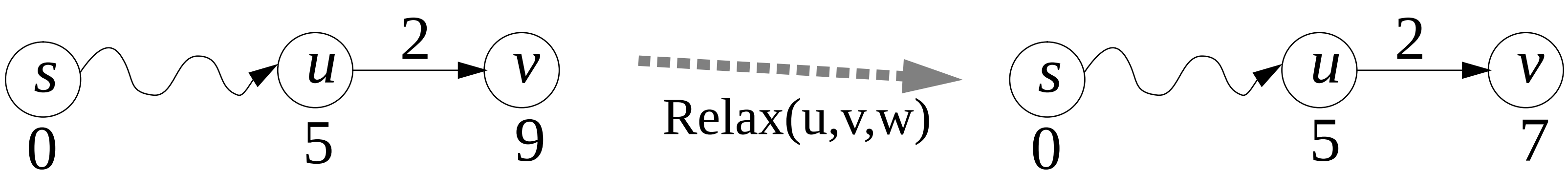
Throughout the algorithms, d[v] is the current best known path from s to v

<u>Relax(u,v,w)</u>

if $d[v] > d[u] + w(u, v)$ then $d[v] = d[u] + w(u, v)$ and $\pi[v]=u$



Example:



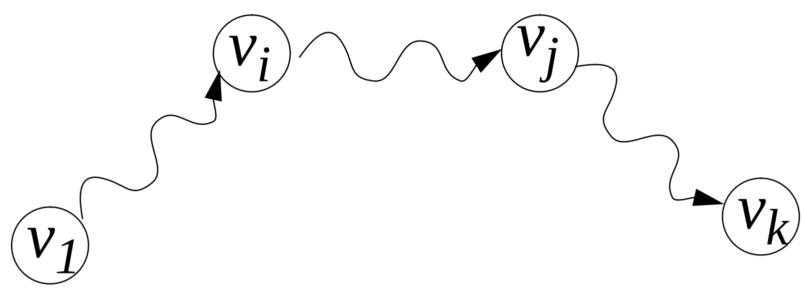
Representing shortest paths

$\pi[v]$=predecessor of v

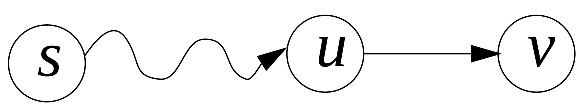
so that the chain of predecessors originating at vertex v runs backwards along a shortest path from s to v

Properties of shortest paths and Relaxation

Lemma: if $p = \langle v_1, v_2, \ldots, v_k \rangle$ is the shortest path from vertex v_1 to vertex v_k, then for any i and j such that $1 \le i \le j \le k$, $p_{ij} = \langle v_i, \ldots, v_j \rangle$ is the shortest path from v_i to v_j



Lemma: For all edges $(u, v) \in E$, we have $\delta(s, v) \le \delta(s, u) + w(u, v)$



TOPIC 19: Single Source Shortest Paths

Shortest Paths

given a graph, we want to compute shortest paths between certain origin and destination vertices

weight of a path = sum of weights of the edges that make up the path

Single source shortest paths: compute shortest paths from origin (source) s to all other vertices

Single-destination shortest paths: compute shortest paths from all vertices to some destination t

All-pairs shortest paths: compute shortest paths between all pairs of vertices

Define: shortest-path distance (or weight)

$\delta(s, v)$ =the minimum weight of a path over all paths from s to v

Issues: negative-weight edges
negative-weight cycle