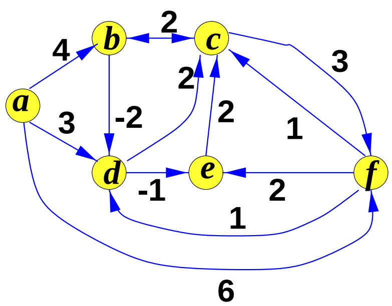
Chapter 25: Single-Source Shortest Paths

The shortest path $\cdots$ the path with the smallest edge-weight sum



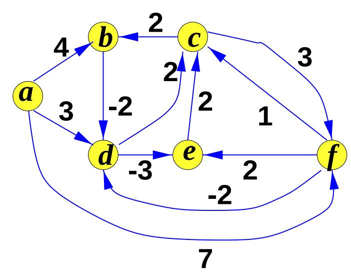
Q1. *What is the length of the path* $\langle a, b, d, c, f \rangle$

$\delta(u,v) \overset{\text{def}}{=}$ the shortest path length

the shortest path problem (SPP) $\cdots$ compute $\delta(u,v)$ for:

1. Single-Source: a fixed u and all v;
2. Single-Destination: a fixed v and all u;
3. Single-Pair: fixed u and v;
4. All-Pair: all u and v.

Negative weight edges can create negative weight cycles, which make the shortest paths undefined



Theorem A

Optimal Substructure Theorem

If $p = \langle v_1, v_2, \cdots, v_k \rangle$ is a shortest path from v_1 to v_k, then for all $i, j, 1 \le i \le j \le k$, $p_{ij} = \langle v_i, \ldots, v_j \rangle$ is a shortest path from v_i to v_j.

1. $\delta(u,v) = \min\{\delta(u,a) + w(a,v) \mid (a,v) \in E\}$.
2. $\delta(u,v) \le \delta(u,a) + w(a,v)$ for any a.

Idea Use variable $d[u,v]$ to compute $\delta(u,v)$

Relaxation with respect to points a, b, c:

$$d[a,c] \leftarrow \min(d[a,c], d[a,b] + w(b,c)).$$

A general strategy

1. Set $d[u,v] = \infty$ for all pairs u, v; repeat below until there is *no update*:
2. *Pick* a, b, c and, **relax** $d[a,c]$ with (b,c):

$$d[a,c] \leftarrow \min(d[a,c], d[a,b] + w(b,c)).$$

3. Repeat (2) until *done*.
4. Output $d[u,v]$ as $\delta(u,v)$.

Strategy for the single-source case (SSSP)

Fix u to a source node s

1. Set $d[v] = \infty$ for all v.
2. *Pick* b, c and, **relax** $d[c]$ with (b, c):

$$d[c] \leftarrow \min(d[c], d[b] + w(b, c)).$$

3. Repeat (2) until *done.*
4. Output $d[v]$ as $\delta(s, v)$.

Dijkstra's algorithm

all weights are nonnegative

$S \stackrel{\text{def}}{=}$ the "finished nodes"; i.e., $\delta(s, v) = d[v]$
$Q \stackrel{\text{def}}{=} V - S$, a priority queue with keys $= d$

1. Set $d[v] = \infty$ for all $v \neq s$, $d[s] = 0$, $S = \emptyset$, and Q to V.
2. While $Q \neq \emptyset$, "extract-min" $u \in Q$, then
   (a) Add u to S.
   (b) For each v with $(u, v) \in E$,

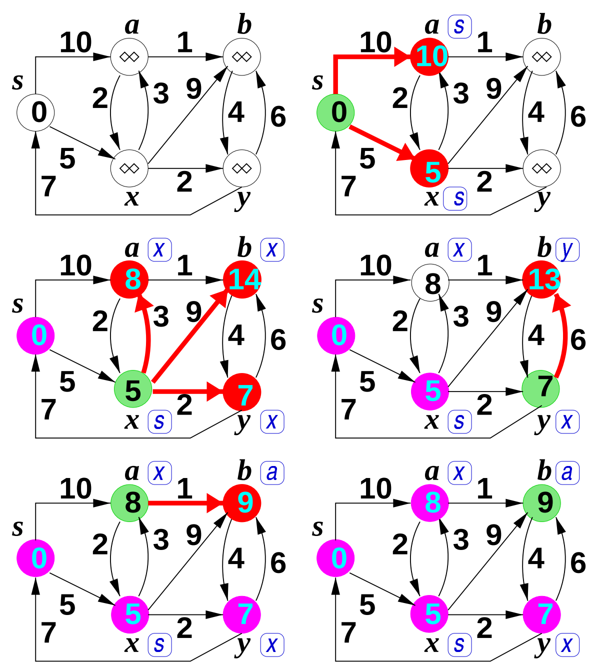
$$d[v] \leftarrow \min(d[v], d[u] + w(u, v)).$$

⋆ Each time $d[b]$ is updated, keep u as the **predecessor** of v in $\pi[b]$.

Q2. *How many times is each edge examined?*

Q3. *How many calls to* EXTRACT-MIN*?*

Q4. *How many calls to* DECREASE-KEY*?*



With Fibonacci heaps, the total running time is $O(E + V \lg V)$.

The Bellman-Ford Algorithm

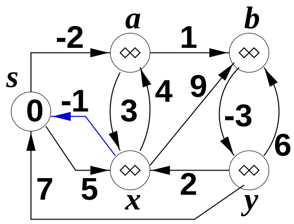
negative weights are allowed
detects existence of negative weight cycles

Repeat $V - 1$ times:

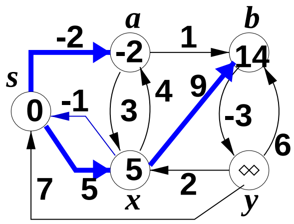
1. For each edge (u, v), *relax with respect to* (u, v)
2. If for some (u, v), $d[v] > d[u] + w(u, v)$, then output "negative weight cycles"

Q5. *How many times is each edge examined?*

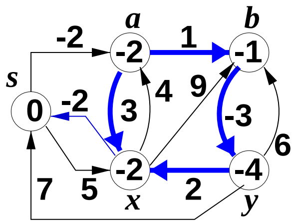
Q6. *What is the running time?*



The edge ordering:
$(a, b), (a, x), (b, y)$,
$(s, a), (s, x)$
$(x, s), (x, a), (x, b)$
$(y, b), (y, s), (y, x)$



The first round.



The second round.
The value of x is first relaxed to 1 by (a, x) then to -2 by (y, x).

$\mu(u)$ = the minimum number of edges in shortest s-u paths; $\mu(u) \leq V - 1$ for all u.

Theorem B If G has no negative weight cycle then for every v, $d[v]$ becomes $\delta(s, v)$ after the $\mu(v)$-th round

Proof By induction on $\mu(v)$.
(Base) $\mu(v) = 0$: trivial because $v = s$
(Induction) Let $\mu(v) = m + 1$ and suppose the claim holds for all u with $\mu(u) \leq m$. Pick an $(m+1)$-edge shortest s-v path p, and let u be the node preceding v. Then $\mu(u) = m$ and

$$\delta(s, v) = \delta(s, u) + w(u, v).$$

So after the m-th round $d[v] = \delta(s, v)$. Thus, in the $(m+1)$st round, $d[u]$ becomes *at most* $\delta(s, u) + w(u, v)$, and this is the smallest it can become. ∎

Theorem C If G has a negative weight cycle then after the $(V - 1)$-st round, for some u, v $d[v] > d[u] + w(u, v)$ holds and the algorithm outputs "negative weight cycle."

Proof Suppose G has a negative weight cycle $\langle v_1, v_2, \ldots, v_k, v_1 \rangle$ whose length is L. Then

$$L = w(v_1, v_2) + \cdots w(v_{k-1}, v_k) + w(v_k, v_1) < 0.$$

Assume, to the contrary, that $d[v] \leq d[u] + w(u, v)$ holds for all u and v after the $(V - 1)$-st round. By our assumption,

$$d[v_i] \leq d[v_{i-1}] + w(v_{i-1}, v_i)$$

for all $i, 1 \leq i \leq k$, where $v_0 = v_k$. Summing these inequialities for all i, we have

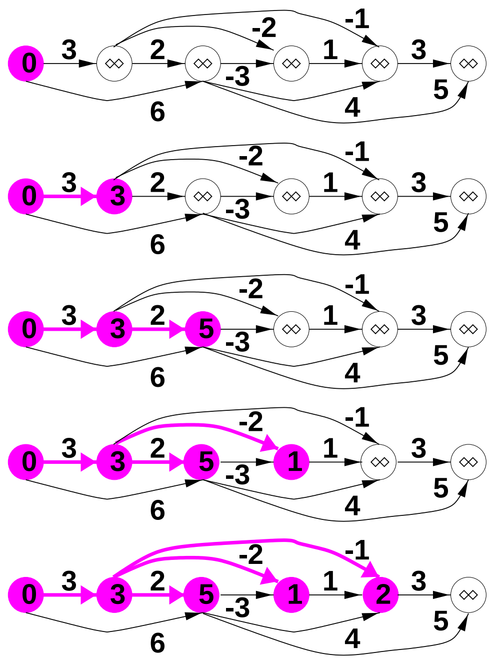
$$\sum_{i=1}^{k} d[v_i] \leq \sum_{i=1}^{k} d[v_i] + L < \sum_{i=1}^{k} d[v_i],$$

a contradiction. ∎

SSSP for a DAG

1. Obtain a topological sort of the nodes.
2. For each $u \neq s$, set $d[u] = \infty$. Set $d[s] = 0$.
3. For each node v in the sorted order, and each u with $(u, v) \in E$, set $d[v] = \min(d[v], d[u] + w(u, v))$.

Q7. *What is the running time?*



Q8. *What is the value of the last node?*