5 November 2021

Given: • a (directed) weighted graph $G=(V,E,\omega)$

$\omega: E \to \mathbb{R}$

• two vertices $s,t \in V$

Want: [the distance / the path] of the shortest path in G from s to t.

• If s, t in diff. conn. comp, $dist(s,t)=\infty$

$path(s,t) = NULL$

More generally: Single Source Shortest Path Problem $\boxed{SSSP}$

Given: $G(V,E,\omega)$ and $s \in V$

Find: SP to every other vertex

note: ① Often, to compute $SP(s,t)$, we need some / most of / all $SSSP(G,s)$

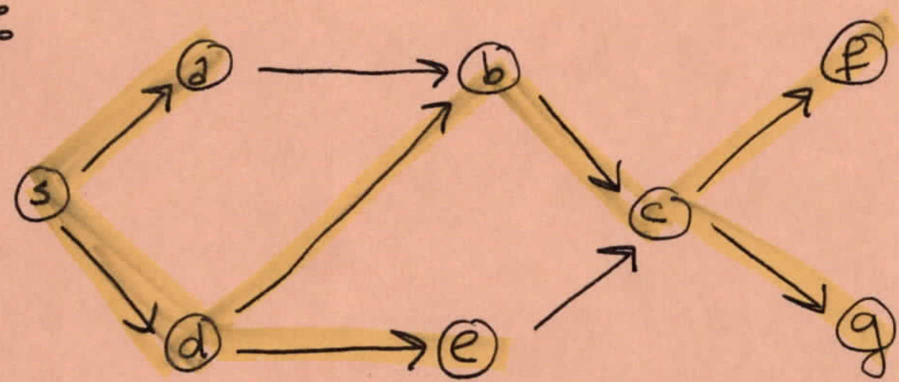
② this can be represented as a tree

$s \xrightarrow{\alpha} a \xrightarrow{\beta} b$ is a SP

$\Rightarrow s \xrightarrow{\alpha} a$ is a SP

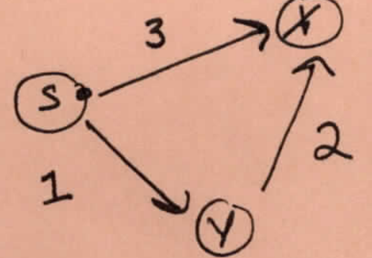
(but might not be unique)

example:



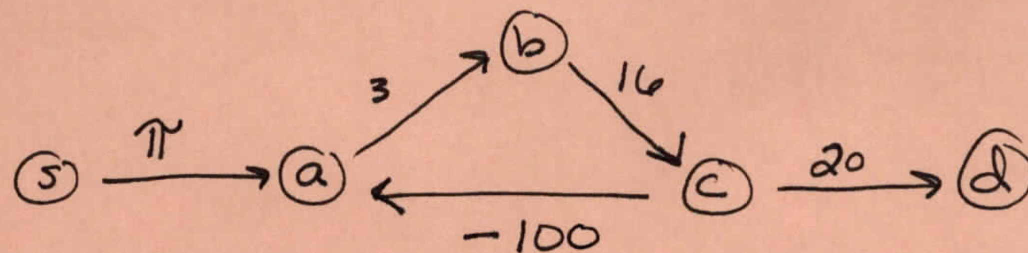
(all unit weights)

note: SP tree need not be unique,
even if edge weights are unique!



} 2 ways to get from s to x w/ total weight 3.

example:



Has no shortest path! $d(s,d) = -\infty$? SP(s,d) = NULL

With neg loops, many "algorithms" will not terminate unless

JE: a) directed graphs + detect these neg. cycles

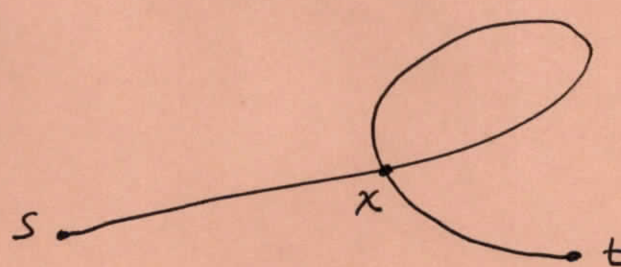
BTF: b) assume we have no negative ~~cycles~~ cycles in undirected graphs

Lemma: Let $G = (V, E, w)$ be a ~~neg~~ weighted graph w/out zero or negative cycles. Then, if p is a shortest path from s to t (denoted $s \overset{p}{\rightsquigarrow} t$), then p is a simple path (i.e. we see each vertex at most once).

Proof: We use a proof by contradiction. Assume p is not simple. Then, p is the concatination of three paths

$$s \overset{p_1}{\rightsquigarrow} x \overset{p_2}{\rightsquigarrow} x \overset{p_3}{\rightsquigarrow} t$$

Picture in mind:



Note that

$$w(p) = w(p_1) + w(p_2) + w(p_3).$$

Also, $p' : s \overset{p_1}{\rightsquigarrow} x \overset{p_3}{\rightsquigarrow} t$ is also a path from s to t and has weight

$$w(p') = w(p_1) + w(p_3) < w(p),$$

since $w(p_2) > 0$. $\blacksquare$

Generic Algo for SSSP

- Given : $G = (V, E, \omega)$, $s \in V$
- Along the way, for each $v \in V$, store:
  1. $dist(v)$ = estimated dist from s to v
  2. $pred(v)$ = predecessor of v in the tentative / estimates SP from s to v
     "the parent" in the SSSP tree

FORD SSSP (G, s)

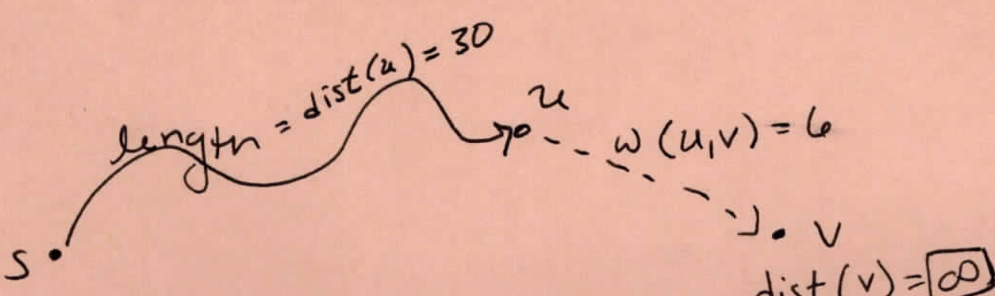
```
for v ∈ V
  | dist(v) ← ∞
  | pred(v) ← NULL
end for
dist(s) ← 0
pred(s) ← NULL
```

initialization of variables

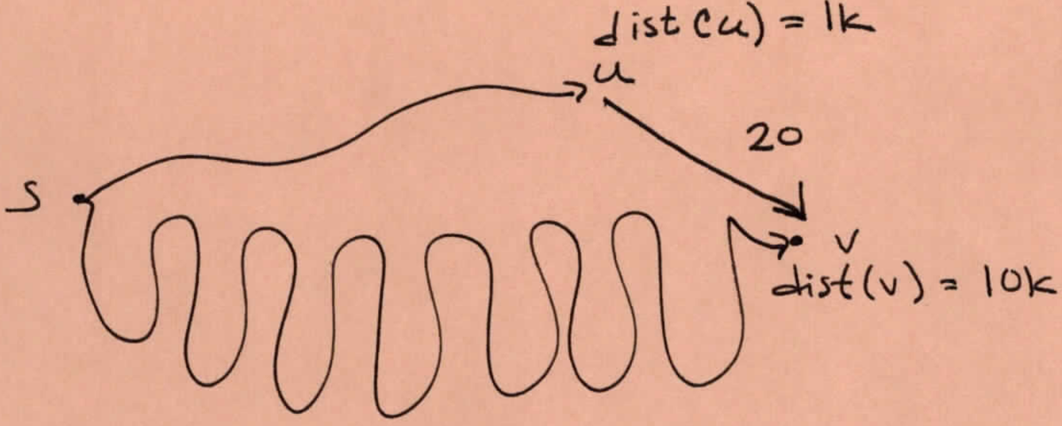
a tense edge

```
while [∃(u,v) ∈ E  s.t.  dist(u) + ω(u,v) < dist(v)]
  | dist(v) ← dist(u) + ω(u,v)
  | pred(v) ← u
end while
return {dist, pred}
```

relax an edge



clearly, this opens up a path!



here, I can take a shortcut