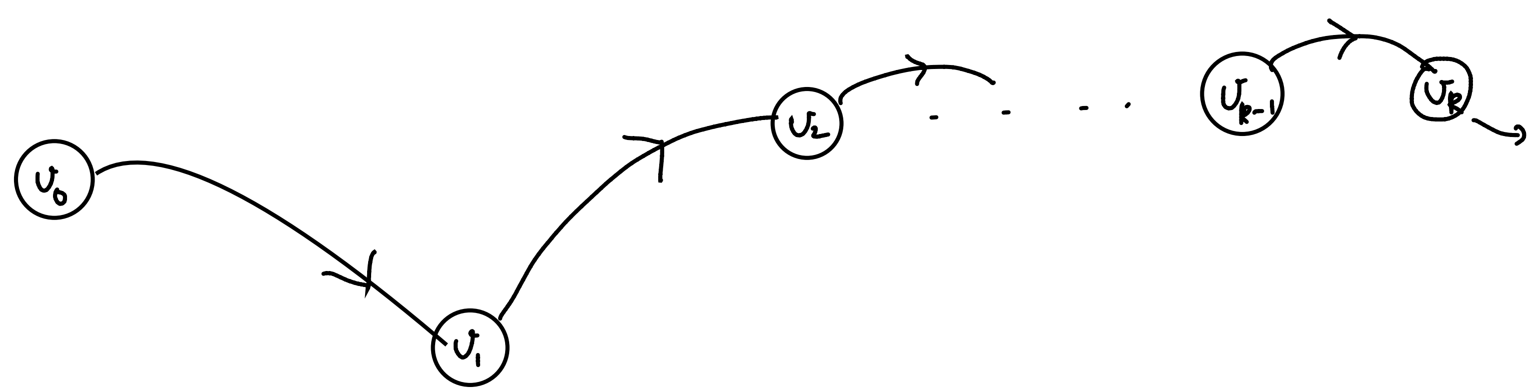
Single - Source Shortest Path

- Directed Graph $G = (V, E)$ $|V|$

- Edge weights $w : E \to \mathbb{R}$

  $w(u, v)$ is weight of edge (u, v)

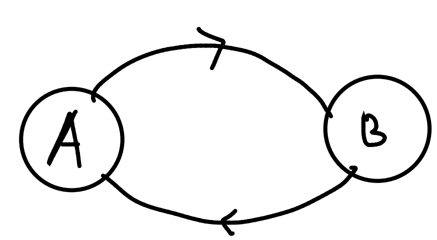
- Path $p = \langle v_0, v_1, \ldots, v_k \rangle$ $v_0 \to v_1 \to v_2 \to \cdots v_{k-1} \to v_k$



k is just a symbol

$v_i = v_j$

Path with cycle



$v_0 = A,\ v_1 = B,\ v_2 = A,\ v_3 = B, \ldots$

$v_k = A$

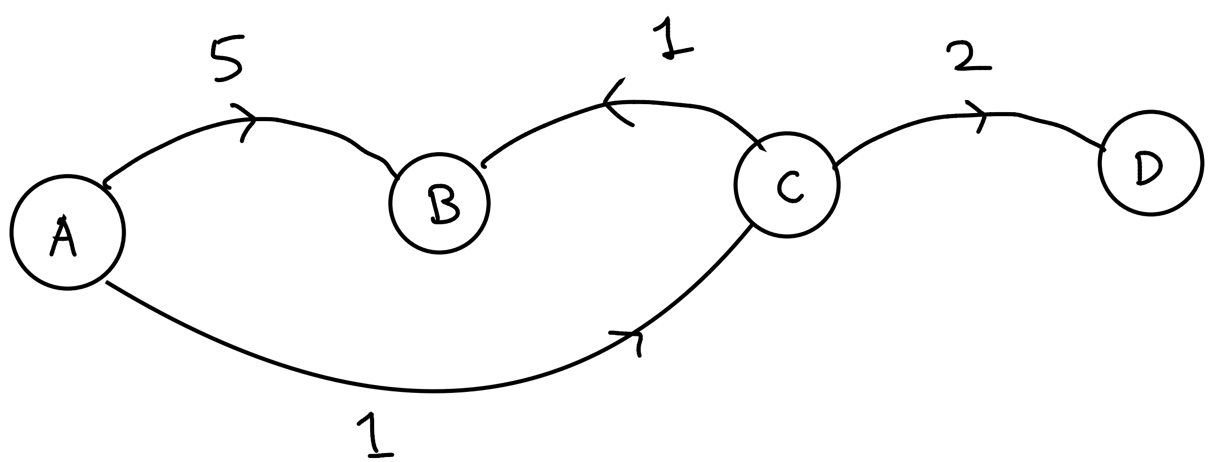
even

- Weight of a path p :

$$w(p) = \sum_{i=1}^{k} w(v_{i-1}, v_i) = \text{summation of weights of individual edges}$$

- Shortest-Path weight between vertices u and v

$$\delta(u,v) = \begin{cases} \min\limits_{u \overset{p}{\rightsquigarrow} v} w(p) & , \text{ if a path exists} \\ \infty & , \text{ if path does not exist} \end{cases}$$



$\delta(A, B) = 2$

$\delta(A, D) = 3$

$\left.\begin{array}{l} \delta(B, A) = \infty \\ \delta(B, C) = \infty \\ \delta(B, D) = \infty \end{array}\right\}$ no path

- We want to find the shortest path from a single source vertex $s \in V$ to each vertex $v \in V$

Special Cases

(i) single destination : $d \in V$ (from each $v \in V$ to $d \in V$)

Flip / Reverse the edges and do single source shortest path

(ii) Between a given pair (u, v)

Properties of Shortest Paths

- Subpaths of shortest paths are shortest paths

$$p = \langle v_1, v_2, \ldots, v_k \rangle$$

$$p_{ij} = \langle v_i, \ldots, v_j \rangle \qquad 1 \le i \le j \le k$$

Proof:

$$v_1 \xrightarrow{p_{1i}} v_i \xrightarrow{p_{ij}} v_j \xrightarrow{p_{jk}} v_k$$

$$w(p) = w(p_{1i}) + w(p_{ij}) + w(p_{jk})$$

say p_{ij} was not shortest path between v_i and v_j, but p'_{ij} is

$$\Downarrow$$

$$w(p_{ij}) > w(p'_{ij})$$

$$\Rightarrow \quad v_1 \xrightarrow{p_{1i}} v_i \xrightarrow{p'_{ij}} v_j \xrightarrow{p_{jk}} v_k$$

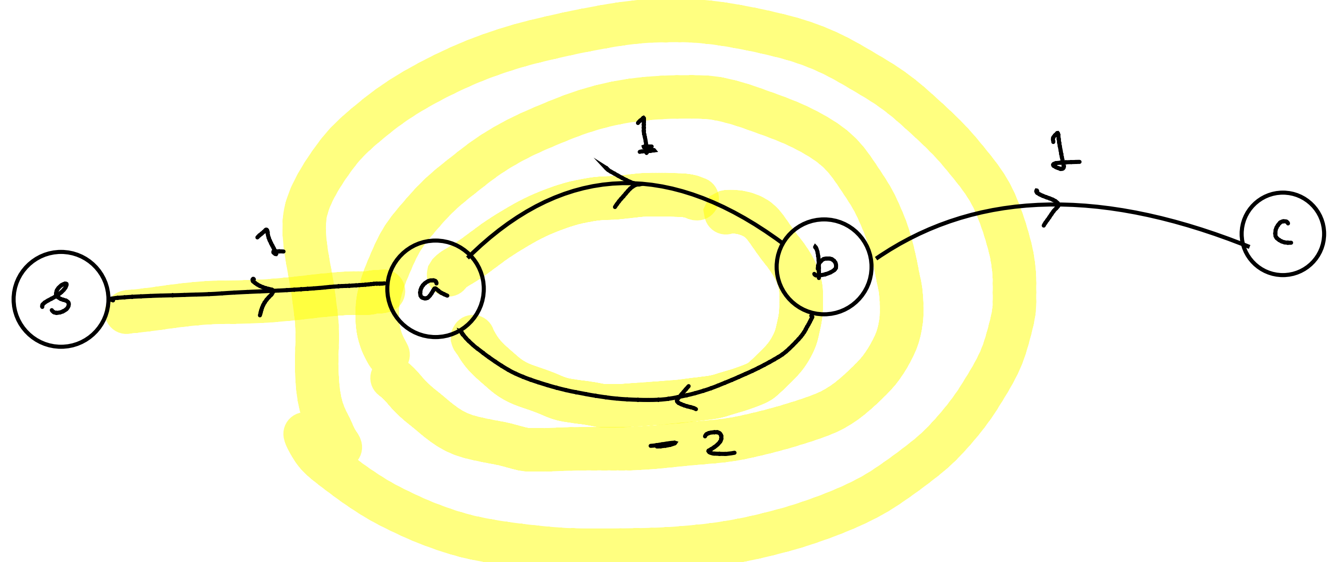
$$\underbrace{w(p_{1i}) + w(p'_{ij}) + w(p_{jk})}_{\text{Shortest Path}} < \underbrace{w(p_{1i}) + w(p_{ij}) + w(p_{jk})}_{\text{not shortest path}}$$

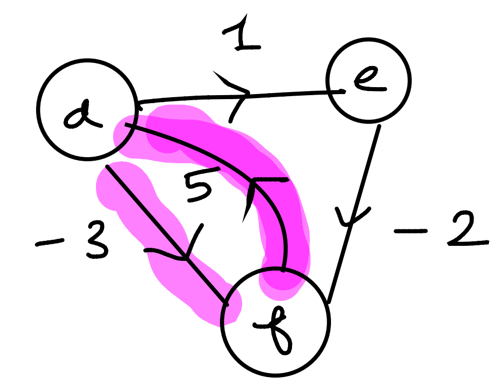
$$\Downarrow$$

Contradiction

- Negative Edges are okay (they are allowed in the graph)
- Negative cycles are an issue

$$\min_{u \overset{p}{\leadsto} v} w(p)$$





$\delta(s, c) = -\infty$

$\delta(s, d) = \infty$

$\delta(d, f) = -3$

$\delta(d, f) = -1$

$\delta(f, d) = 5$

- Shortest Paths contain no +ve cycles (by removing this +ve cycle we can always produce a shorter path)

- $|V|$ distinct vertices

$\Downarrow$

$|V|-1$ distinct edges in an acyclic path

- Predecessor $\pi[v]$ = another vertex

= NIL

Important

PRINT-PATH $(\underbrace{G, s, v}^{\text{args}})$

if $v = s$

then print s

else if $\pi[v]$ = NIL

then print "$\underbrace{\text{no path from}}_{\text{text}}$" $\underbrace{s}_{\text{arg}}$ "to" $\underbrace{v}_{\text{arg}}$ "exists"

else PRINT-PATH $(G, s, \pi[v])$

print v

- Predecessor Graph

$$G_\pi = (V_\pi, E_\pi)$$

$$V_\pi = \{ v \in V : \pi[v] \neq NIL \} \cup \{ s \}$$

does not contain vertices with no path

$$E_\pi = \{ (\pi[v], v) \in E : v \in V_\pi - \{s\} \}$$

only those edges involved in shortest path

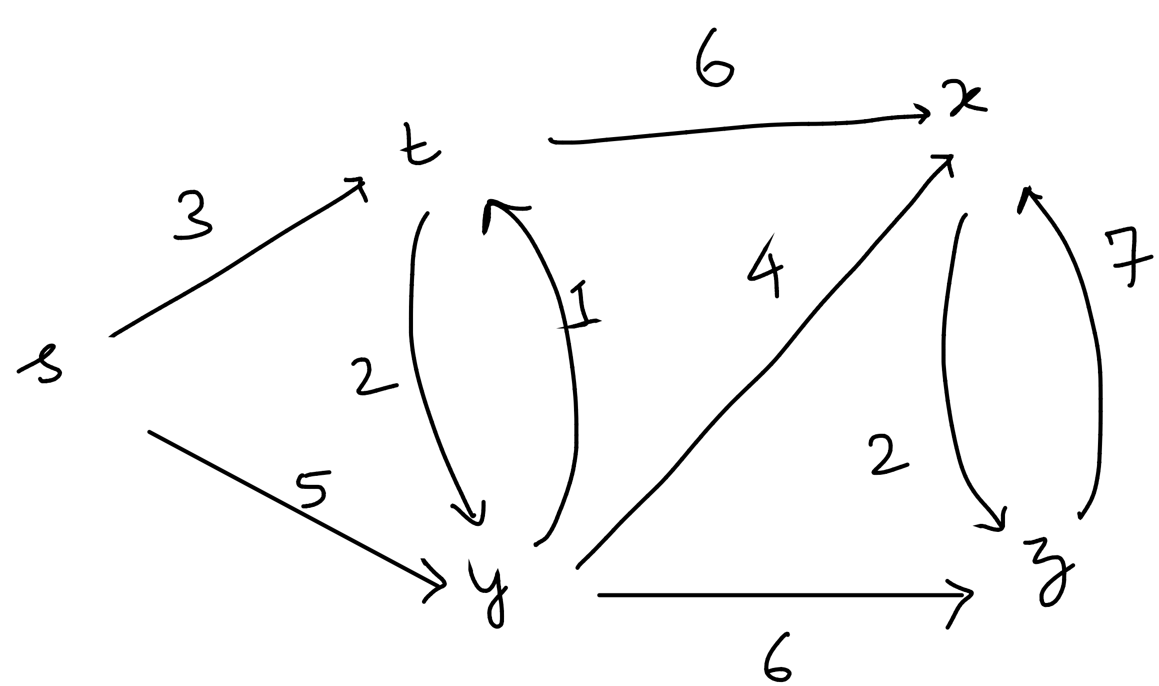
- Ideally, at termination of our algorithms, we want G_π to "Shortest-Path-Tree"

- Shortest Path Tree : $G' = (V', E')$

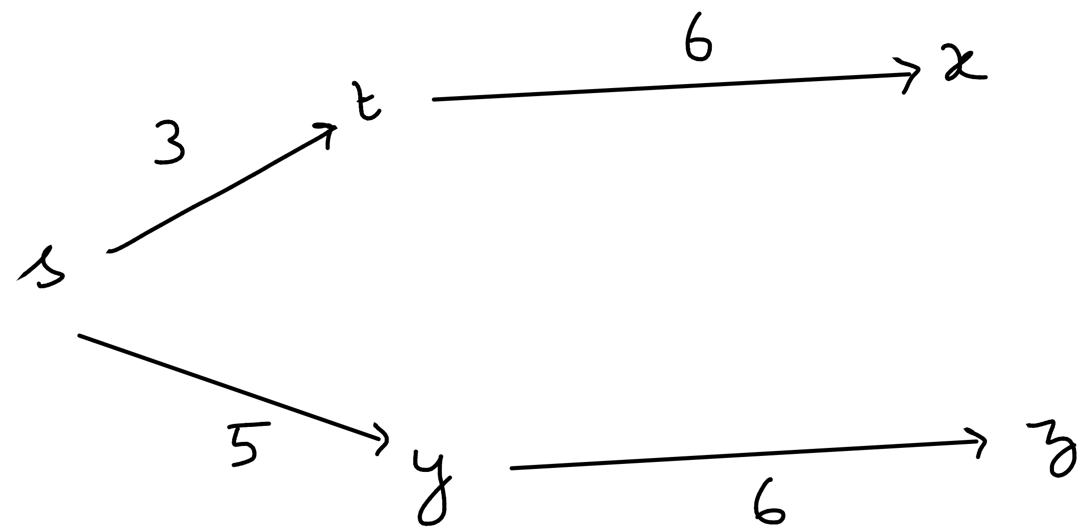
  - $V' \subseteq V$ is the set of vertices reachable from s in G
  - G' is a tree rooted at s
  - for all $v \in V'$, the simple path from s to v in G' is the shortest path from s to v in G

Shortest Path Tree Need Not be unique

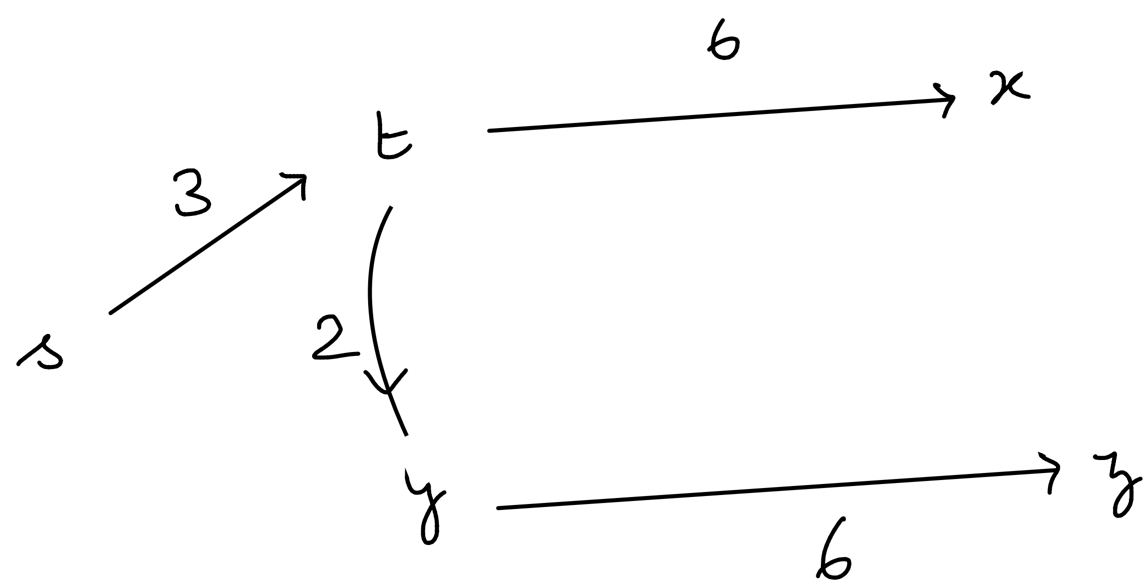
$G = (V, E)$



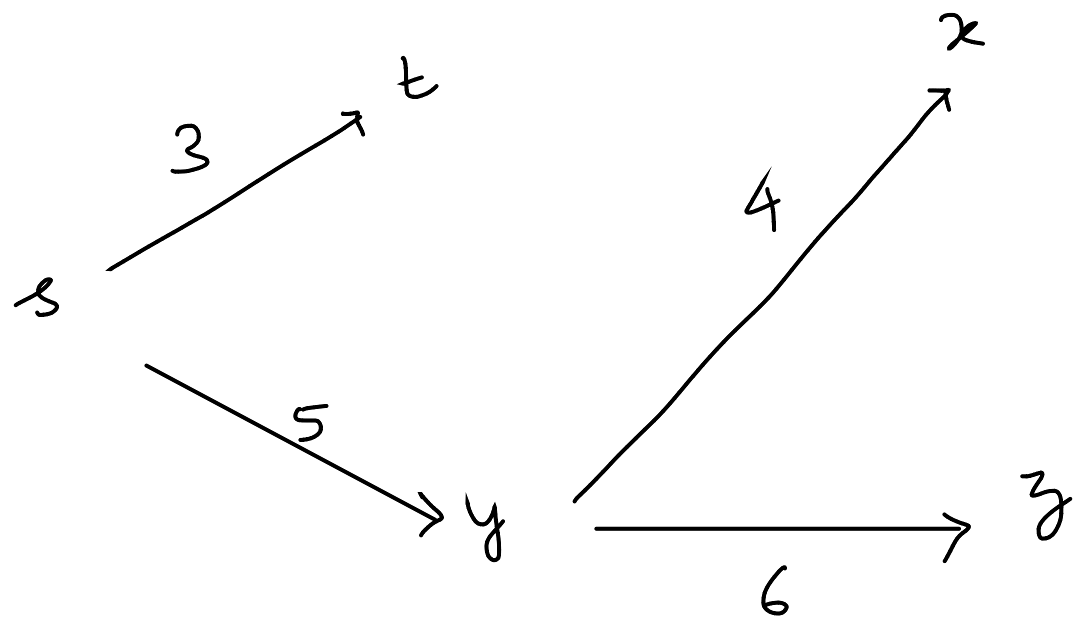
$G' = (V', E')$



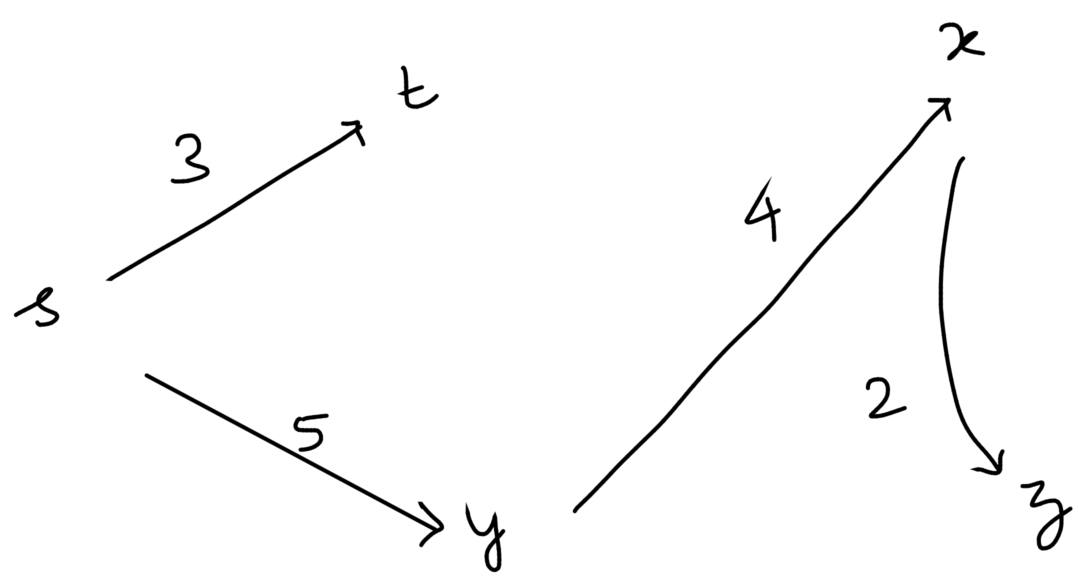
$G'' = (V'', E'')$



$G''' = (V''', E''')$



$G = (V, E)$



- Relaxation

$d[v]$: upper bound on weight of shortest path to v (from s is impelicit)

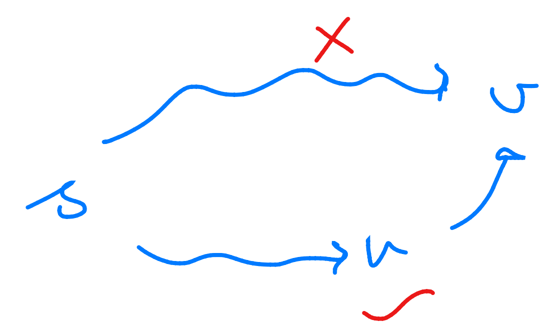
INIT - SINGLE - SOURCE (G, s)

for each vertex $v \in V(G)$

do $d[v] \leftarrow \infty$

$\pi[v] \leftarrow$ NIL

$d[s] \leftarrow 0$



- Relax (u, v, w)

if $d[v] > d[u] + w(u,v)$

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

$\pi[v] \leftarrow u$

Basically to reach v it seems a "better" idea to reach to u first and take edge (u, v)