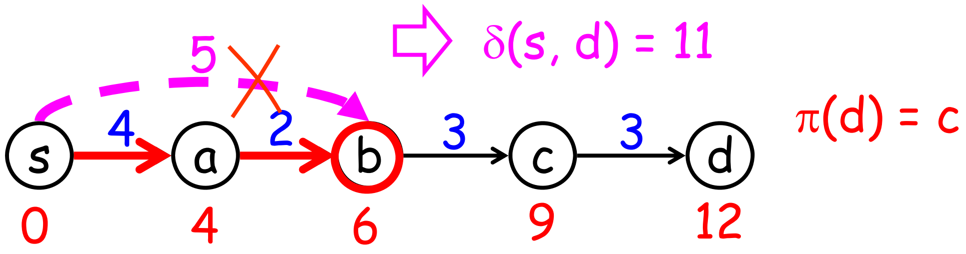
Main Idea ----- 1

If



is a shortest path from s to d

Then

(i) all subpaths are shortest **optimal substructure !**

(ii) After $\delta(s, \pi(v))$ is known,
we can get $\delta(s, v)$ by Relax($\pi(v)$, v, w)

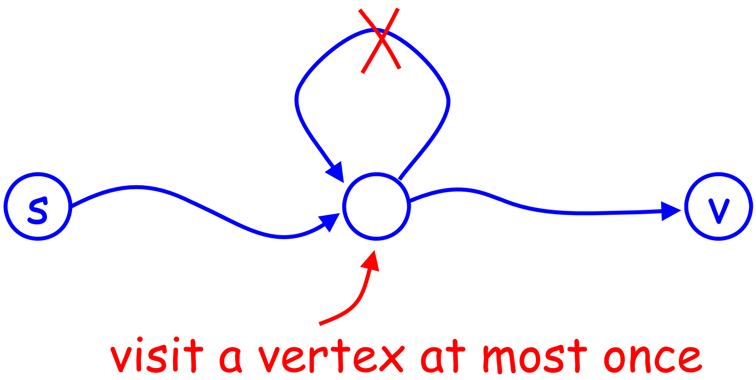
e.g. After $\delta(s, c) = 9$ is known,
we have $\delta(s, d) = 9 + w(c, d) = 12$ **Relax(c, d, w)**

Main Idea ----- 2

If G contains no negative cycles,

(i) every shortest path is a simple path

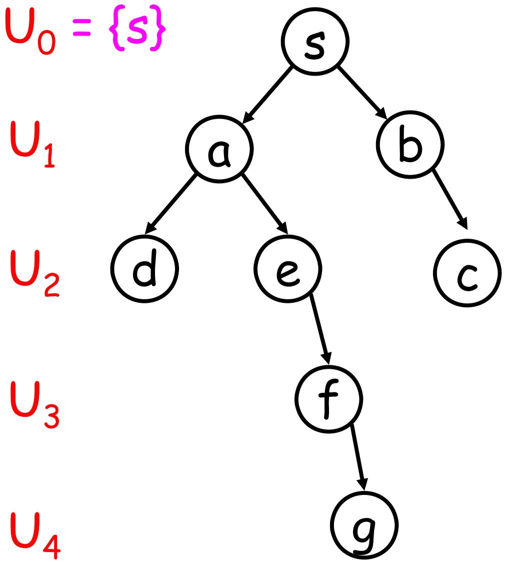
(ii) every shortest path has at most n – 1 edges



(For ease of discussion, assume that there are no 0-cycles)

Main Idea: Bellman-Ford
(no negative cycles)

shortest path tree



* U_i: vertices whose shortest paths having i edges

* $U_0 \xrightarrow{\text{phase 1}} U_1 \xrightarrow{\text{phase 2}} U_2 \longrightarrow \cdots$

ok ok ok

main idea 1 - correctness

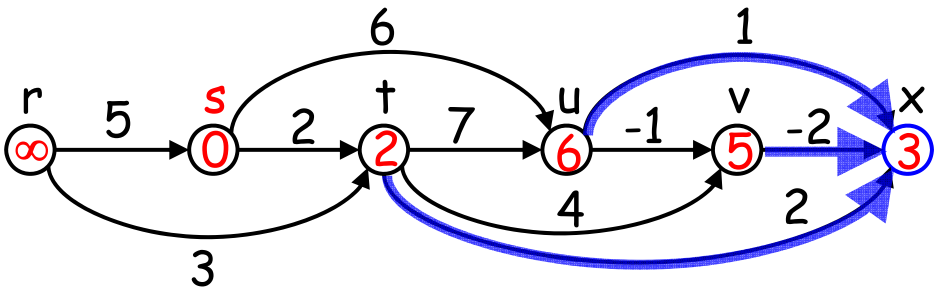
* A simple path has at most n – 1 edges

⇨ $U_n = U_{n+1} = U_{n+2} = \ldots = \varnothing$

⇨ n – 1 phases is sufficient!

main idea 2 - time complexity

Traditional approach: DP (See 15-14a)



$$\begin{cases} d(s) = 0 \\ d(v) = \min\limits_{(u,v) \in E} \{d(u) + w(u,v)\} \end{cases}$$

bottom-up computation
(left-to-right)

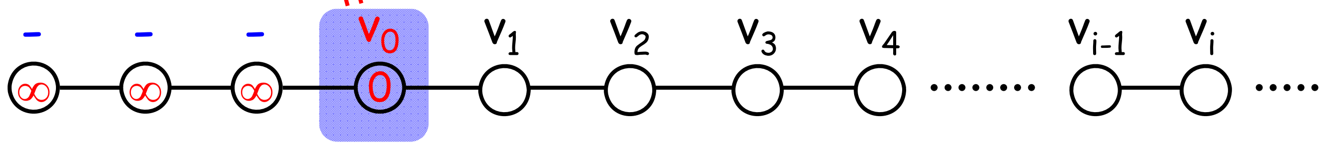
$\pi(x) \in \{t, u, v\}$

$$d(x) = \begin{cases} d(t) + 2, \\ d(u) + 1, \\ d(v) + (-2) \end{cases}$$

DP: 有答案的存起來等別人問 (t, u, v等x來問答案)

24.2: 有答案的主動去修正有需要的人 (t, u, v主動用答案修正x)

arrange vertices in topological order



* all edges are from left to right →

* $\pi(v_i)$ is one of $v_0, v_1, v_2, \ldots, v_{i-1}$ (or NIL)

* Once $v_0, v_1, v_2, \ldots, v_{i-1}$ ok ⇨ v_i ok!

* Initially, $d(v_0)$ is correct

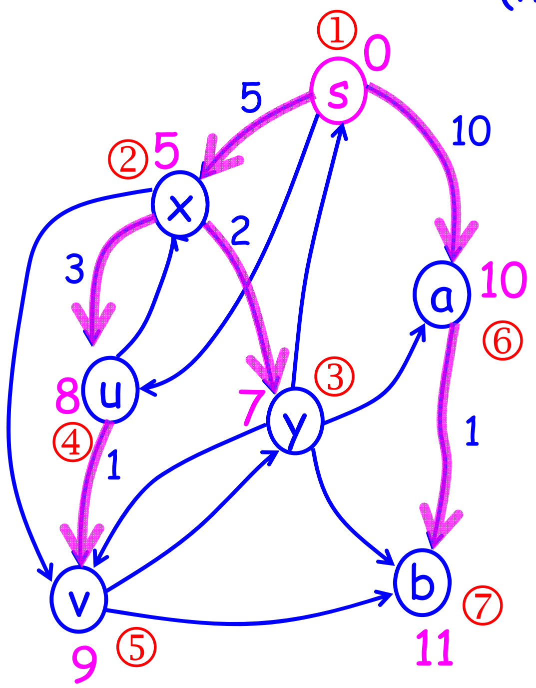
  v_0 does "relax" with correct $d(v_0)$ ⇨ $d(v_1)$ is correct

⇨ v_1 does "relax" with correct $d(v_1)$ ⇨ $d(v_2)$ is correct

⇨ v_2 does "relax" with correct $d(v_2)$ ⇨ $d(v_3)$ is correct

⇨ • • • all $d(v_i)$ are correct (by induction)

Main Idea: Dijkstra
(no negative edge)



No negative edge → $\delta(v) > \delta(\pi(v))$

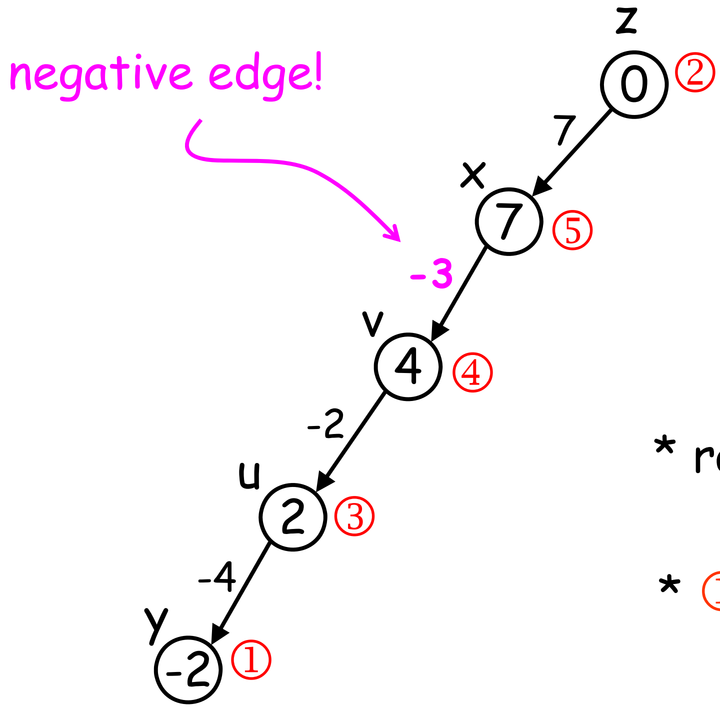
⇨ rank(v) > rank(π(v))

⇨ Once ① ② ③ • • • ⓚ ok,
(k+1) can be computed.

⇨ ① → ② → ③ → ④ → • • •
ok ok ok ok

必 然 是 s

Why all weights should be nonnegative?



(shortest path tree of 24-5 Fig.)

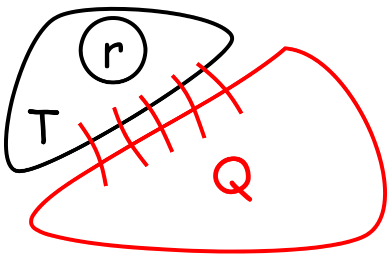
Dijkstra's idea :

$\cancel{\mathrm{rank}(v) > \mathrm{rank}(\pi(v))}$

* $\mathrm{rank}(v) < \mathrm{rank}(\pi(v))$
④ ⑤

* ① ② ③ ok ⇨ v not ok
④

Prim's MST



key[v] : shortest edge to T

$\pi[v]$: nearest vertex in T

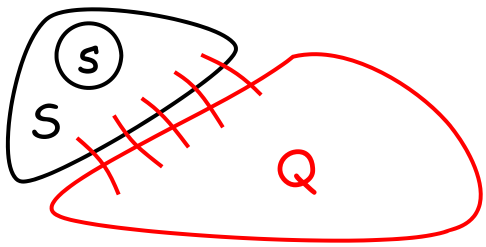
u ← ExtractMin(Q)

$T \leftarrow T \cup \{u\}$

reduce key[·] of Adj(u)

(decrease-key)

Dijkstra's shortest path



d[v] : known shortest distance to s

$\pi[v]$: current predecessor

u ← ExtractMin(Q)

$S \leftarrow S \cup \{u\}$

relax d[·] of Adj(u)

(decrease-key)