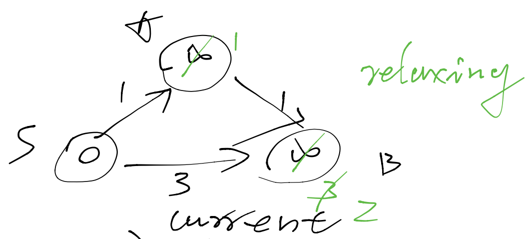
Dijkstra

□ Review




- $d[v]$ : length of the shortest path from $s$ to $v$
- $\delta(s,v)$ : length of a shortest path
- $\pi[v]$ : predecessor of $v$ on the shortest path from $s$ to $v$
- Relax $(u, v, w)$ :

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

  $d[v] = d[u] + w(u,v)$

  $\pi[v] = u$

○ Lemma :

The relexation operation maintain the invarient that $d[v] \geq \delta(s, v)$ for all $v \in V$ ( relaxzation is " safe " )

Proof :

By induction on the number of steps

By induction $d[u] \geq \delta(s, u)$

By triangle-inequality: $\delta(s,v) \le \delta(s,u) + \delta(u,v)$

$$\begin{aligned}\delta(s,v) &\le d[u] + \delta(u,v)\\ &\le d[u] + w(u,v)\\ &= d[v]\end{aligned}$$

□ Directed acyclic graphs (DAGs)

Can't have negative cycles, but allow negative edges

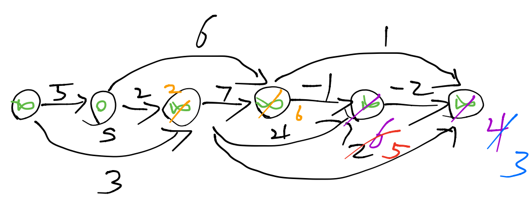
1) Topologically sort the DAG path from u to v
Implies that u is before v in the ordering

2) One pass over vertices in topologically sorted order, relax each edge that leaves the vertex.

$O(V+E)$ time

e.g.

DAG.



□ Dijkstra Algorithm

( Doesn't

Dijkstra (G, W, s)

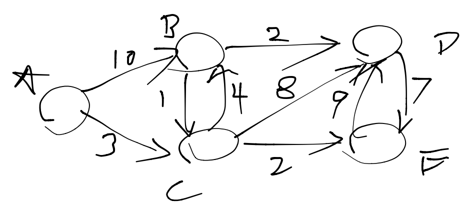
Initialize (G, s), $S \leftarrow \phi$ $Q \leftarrow V[G]$ ↙ priority Queue while $Q \neq \emptyset$

$u \leftarrow$ EXTRACT-MIN (Q) // delete u from Q

$S = S \cup \{u\}$

for each vertex $v \in Adj[u]$

Relax (u, v, w)




$S = \{\}$ $\quad Q = \{A, B, C, D, E\}$

$0, \infty, \infty, \infty, \infty$

$S = \{A\}$ $\quad 0, 10, 3, \infty, \infty$

$S = \{A, C\}$ $\quad 0, 7, 3, 11, 5$

$S = \{A, C, E\}$ $\quad 0, 7, 3, 11, 5$

- Cost:

$\Theta(V)$ inserts into the priority Queue

$\Theta(V)$ extract_min ops

$\Theta(E)$ Decrease-key ops

Arrays

$\Theta(V)$ extract-min

$\Theta(1)$ decrease key

total: $\Theta(V \cdot V + E) = \Theta(V^2)$

Binary min_heap

$\Theta(\lg V)$ for extract-min

$\Theta(\lg V)$ for decrease key

$\Theta(V\lg V + E\lg V)$