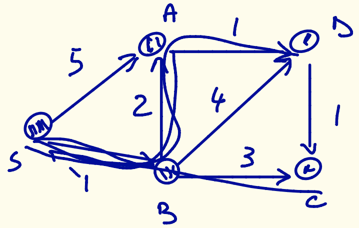
SHORTEST PATH - CONTINUED

Dijkstra's Algorithm

prec = array indexed by vertices initialized to NIL
dist = array indexed by vertices initialized to ∞
Q = priority queue of vertices indexed by dist[·]

dist[s] = 0
For each v: Q.insert(v)
while Q is not empty
    v = Q.deletemin()
    for each w neighbor of v:
        if dist[w] > dist[v] + $\ell(v,w)$:
            dist[w] = dist[v] + $\ell(v,w)$
            Q.decreasekey(w)
            prec[w] = v



$Q = \{$

$\text{dist}[\ 0 \quad 3 \quad 1 \quad 4 \quad 4\ ]$

$\phantom{\text{dist}[\ } S \quad A \quad B \quad C \quad D$

$\text{prec}[\ \perp \quad \perp \quad \perp \quad \perp \quad \perp\ ]$

Properties:

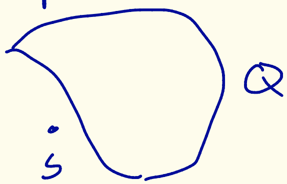
At the end of each iterations

A nodes outside Q have dist[v] = length of shortest path from s to v

B every node has dist[v] = length of shortest path from s to v that uses only nodes not in Q as intermediate steps

---

First step

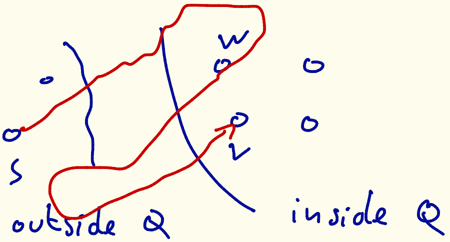


$dist[s] = 0$

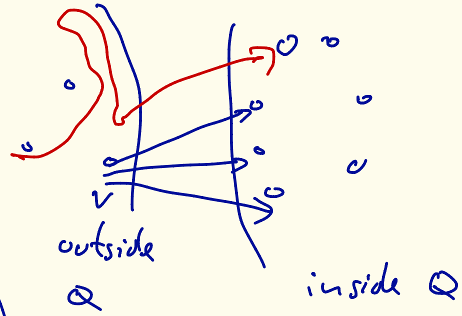
$dist[v] = \ell(s, v)$ if exists

$= \infty$ if no edge from s to v

time t



time $t+1$



$v = Q.\text{deletemin}()$

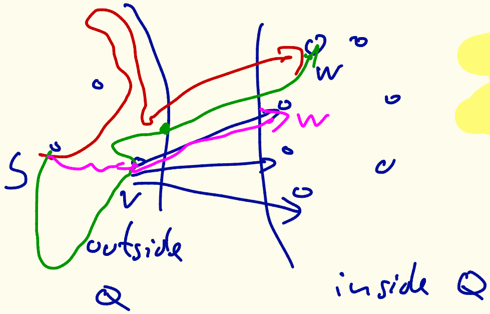
Prop. A is true at step $t+1$

take any path $s \to v$

1. if path has only nodes not in Q as intermediate steps, length $\geq$ dist[v] [from B at step t]

2. if path passes through some vertex w in Q, Part of path $s \to w$ only has vertices $\overline{Q}$ as intermediate steps
   Part of path $s \to w$ is of length $\geq$ dist[w]
   dist[w] $\geq$ dist[v]

   length of path from $s \to v$
   $\geq$ part of path from s to w
   $\geq$ dist[w] $\geq$ dist[v] [from B at step t]



Prop B is true at step $t+1$

take any path from s to w that has only nodes $\overline{Q}$ as intermediate steps

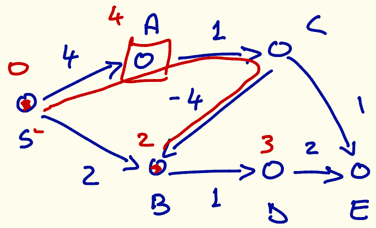
(1) if path does not contain v
length path $\geq$ dist[w]
by inductive assumption

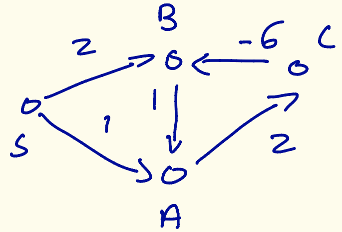
(2) if path uses v not in 2nd-to-last step it is not shortest

(3) if path uses v as the second-to-last step

$$\text{dist}[w] \leq \text{dist}[v] + \ell(v,w)$$
$$\leq \text{length of path}$$

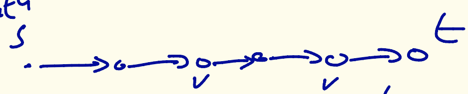
Negative weight edges





Suppose G is a directed weighted graph with <u>no</u> negative cycle
Then for every s, t if there is a path from s to t then there is a shortest path

Take any path



suppose some vertex is repeated

remove $v \to v$ cycle

path of length $\leq$ before

every path $\geq$ shortest path with no repeating vertices

Bellman-Ford

dist = array indexed by V initialized to ∞

prec = array indexed by V initialized to $\perp$

dist [S] = 0

for $\ell = 1$ to $|V|-1$:

for each v in $V - \{s\}$: $\quad (|V|-1) \cdot \overbrace{\sum_{v \in V} \text{indegree}(v)}^{|E|}$

for each edge (u,v) :

if dist[u] + $\ell(u,v)$ < dist[v] :

dist[v] = dist[u] + $\ell(u,v)$

prec[v] = u

Running time $O(|V| \cdot |E|)$

Correctness : At step ℓ of outer for

for every v $\quad$ dist[v] $\leq$ length of shortest path from s to v that uses $\leq \ell$ edges