Proof of Dijkstra's Algorithm – A Deeper Dive

This deck unpacks some of the details in the proof of Dijkstra's algorithm given in class

Shortest paths in a graph with cycles and nonnegative weights

Def: $\delta(s, v)$ = minimum distance from s to v.

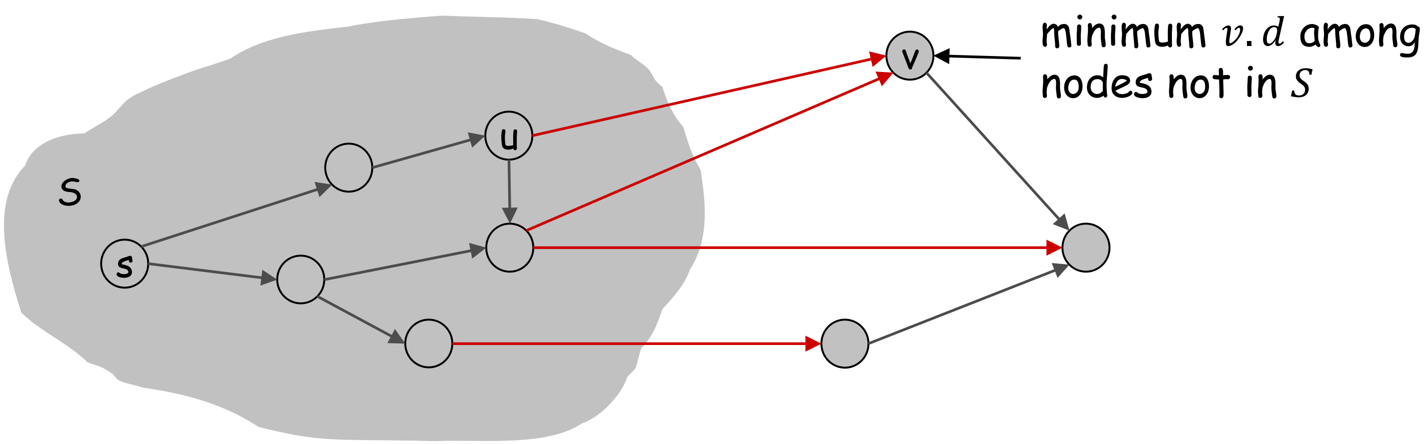
Challenge: The same recurrence holds, but there is no obvious order in which to compute the recurrence if the graph has cycles.

Dijkstra's algorithm.

- Maintain a set of explored nodes S for which we **know** $u.d = \delta(s, u)$.
  Initialize $S = \{s\}, s.d = 0$, $v.d = \infty$

Key lemma: Suppose $u.d = \delta(s, u)$ for all $u \in S$, and all edges leaving S have been relaxed. Then $v.d = \delta(s, v)$, where v is the vertex with minimum $v.d$ in $V - S$.

So this v can be added to S, and we then repeat



Dijkstra's Algorithm

Dijkstra(G,s):

for each $v \in V$ **do**

    $v.d \leftarrow \infty$, $v.p \leftarrow nil$, $v.color \leftarrow white$

$s.d \leftarrow 0$

create a min priority queue Q **on** V **with** d **as key**

while $Q \neq \emptyset$

    $u \leftarrow$ **Extract-Min(**Q**)**

    $u.color \leftarrow black$

    for each $v \in Adj[u]$ **do**

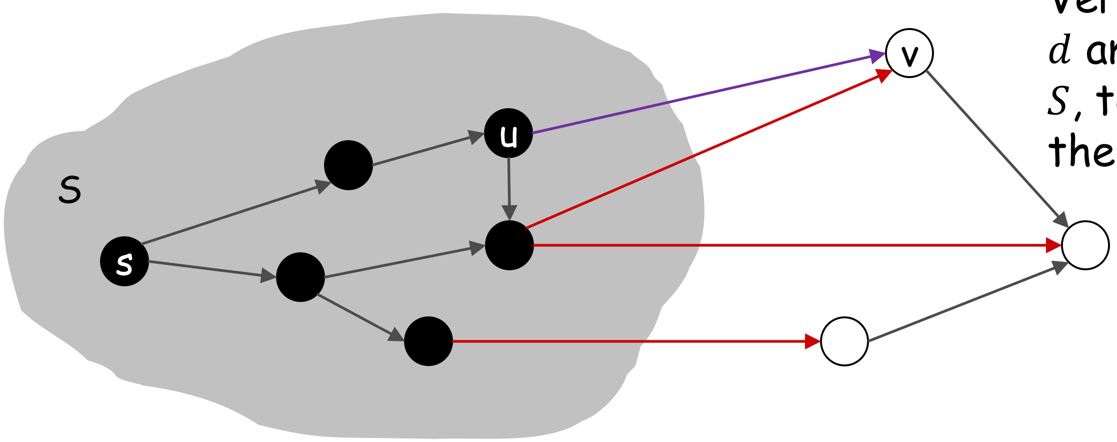
        if $v.color = white$ **and** $u.d + w(u,v) < v.d$ **then**

            $v.p \leftarrow u$

            $v.d \leftarrow u.d + w(u,v)$

            Decrease-Key($Q, v, v.d$**)**

Relax all(u,v**) when u gets added to** S

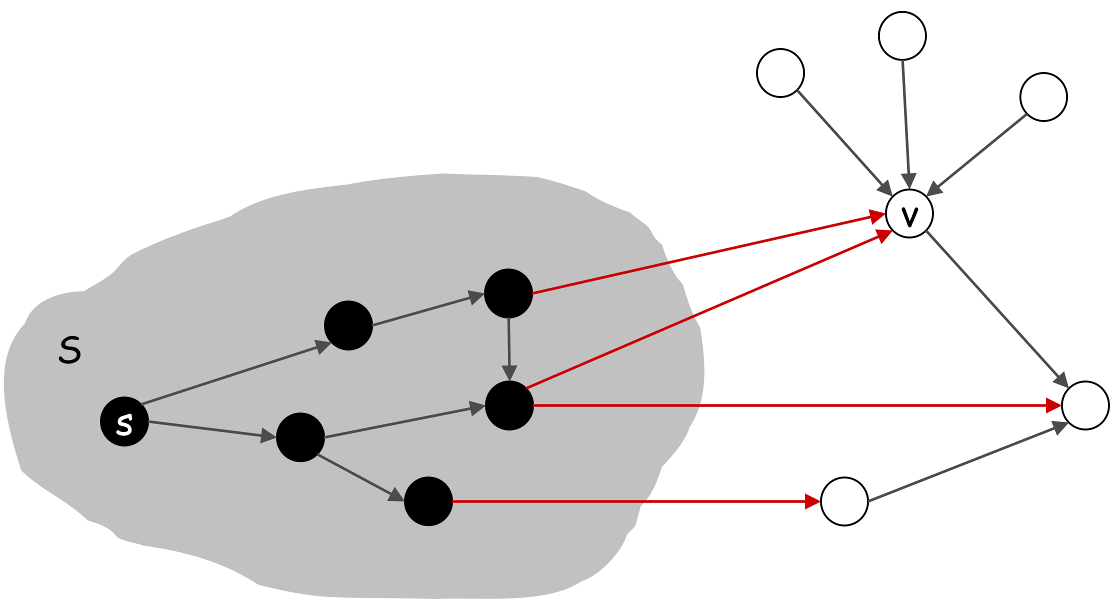


Vertex with minimum d among nodes not in S, to be added to S in the next round

Dijkstra's Algorithm: Correctness

Lemma. Suppose $u.d = \delta(s,u)$ for all $u \in S$, and all edges leaving S have been relaxed. Then $v.d = \delta(s,v)$, where v is the vertex with minimum $v.d$ in $V - S$.

- Note that the red edges were relaxed when their sources were added to S (turning them black), and they will not be relaxed any more in the future (discussed in the DAG case)
- Note that $v.d$ starts $= \infty$. Whenever $v.d$ is updated, it's because a path with a shorter distance than the current $v.d$ was found. If $v.d$ gets reduced further in later steps, the relaxation must come from some edge that is currently outside of S (non-red edge from a white vertex to v).

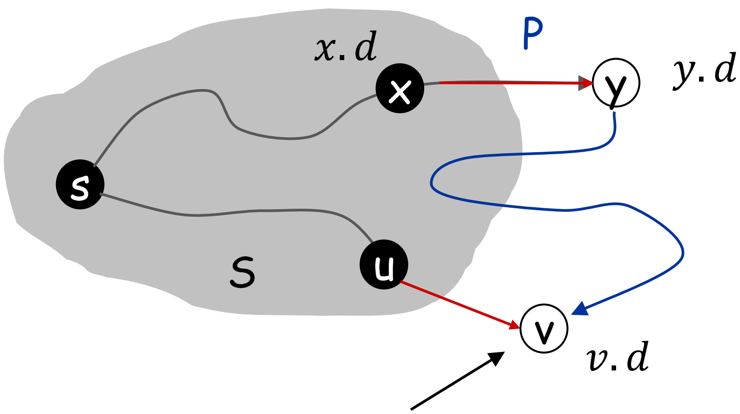


Dijkstra's Algorithm: Correctness

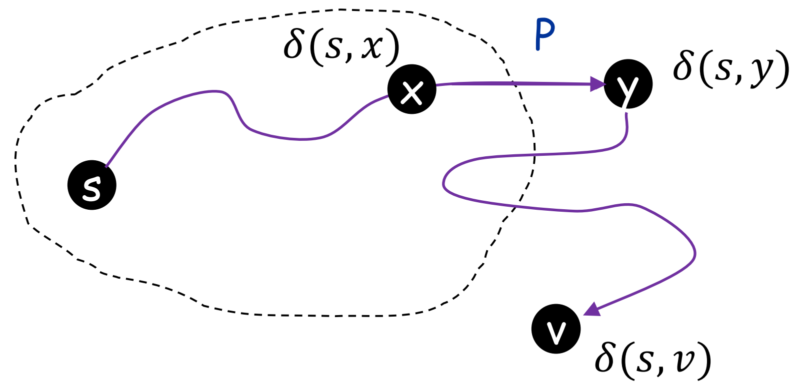
Lemma. Suppose $u.d = \delta(s,u)$ for all $u \in S$, and all edges leaving S have been relaxed. Then $v.d = \delta(s,v)$, where v is the vertex with minimum $v.d$ in $V - S$.

Pf. (by contradiction)

- Note that $v.d$ starts $= \infty$. Whenever $v.d$ is updated, it's because a path with distance $v.d$ was found. So always have $v.d \geq \delta(s,v)$.
  Thus if $v.d \neq \delta(s,v)$ then $v.d > \delta(s,v)$ --- ❶.



$v.d$ is the smallest among nodes not in S



A shortest path from s to from v

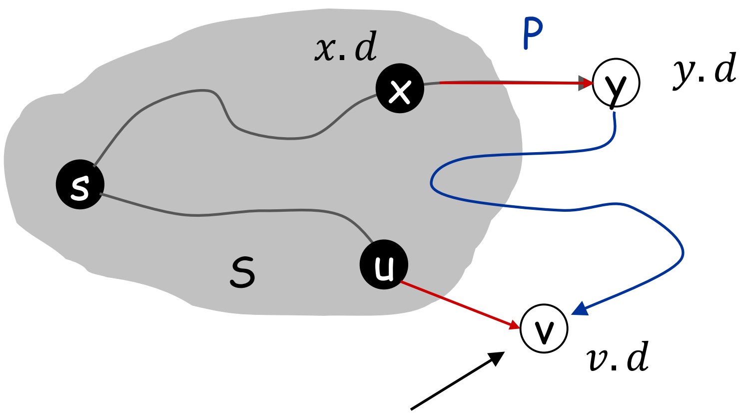
- Consider a shortest path P from s to v, shown in the right graph and mapped to the left
  - Suppose $x \to y$ is the first edge on P that takes P out of S.
  - *By our definition, it is possible that $y = v$ and $x = u$*

Dijkstra's Algorithm: Correctness

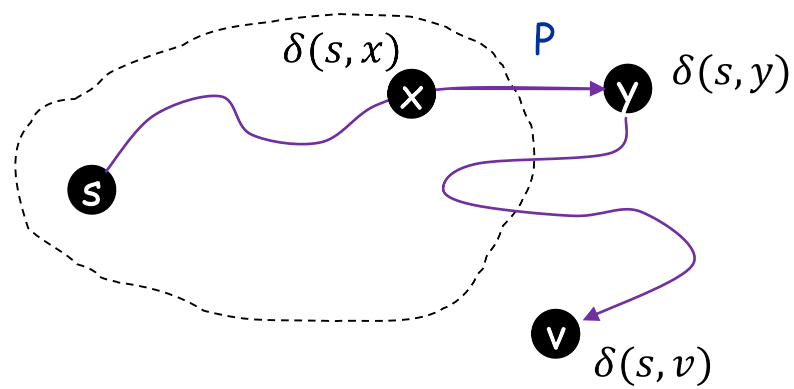
Lemma. Suppose $u.d = \delta(s,u)$ for all $u \in S$, and all edges leaving S have been relaxed. Then $v.d = \delta(s,v)$, where v is the vertex with minimum $v.d$ in $V - S$.

Pf. (by contradiction)

- Note that $v.d$ starts $= \infty$. Whenever $v.d$ is updated, it's because a path with distance $v.d$ was found. So always have $v.d \geq \delta(s,v)$.
  Thus if $v.d \neq \delta(s,v)$ then $v.d > \delta(s,v)$ --- ❶.



$v.d$ is the smallest among nodes not in S



A shortest path from s to from v

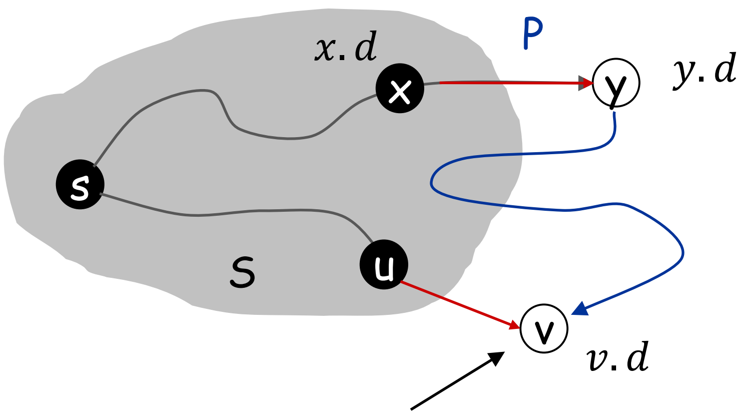
- Consider a shortest path P from s to v, shown in the right graph and mapped to the left
  - P is shortest path, its subpath $(s, \dots, y)$ and $(y, \dots, v)$ must also be the shortest. According to the cut and paste argument, $\delta(s,v) = \delta(s,y) + \delta(y,v)$ --- ❷
  - $\delta(s,v) \geq \delta(s,y)$, assuming nonnegative weights --- ❸

Dijkstra's Algorithm: Correctness

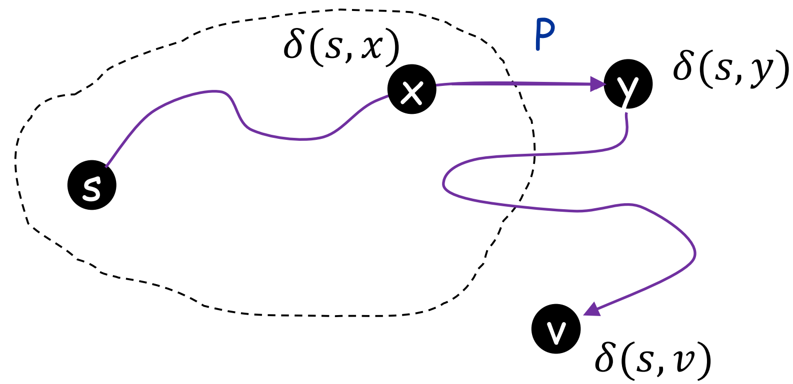
Lemma. Suppose $u.d = \delta(s,u)$ for all $u \in S$, and all edges leaving S have been relaxed. Then $v.d = \delta(s,v)$, where v is the vertex with minimum $v.d$ in $V - S$.

Pf. (by contradiction)

- Note that $v.d$ starts $= \infty$. Whenever $v.d$ is updated, it's because a path with distance $v.d$ was found. So always have $v.d \geq \delta(s,v)$.
  Thus if $v.d \neq \delta(s,v)$ then $v.d > \delta(s,v)$ --- ❶.



$v.d$ is the smallest among nodes not in S



A shortest path from s to from v

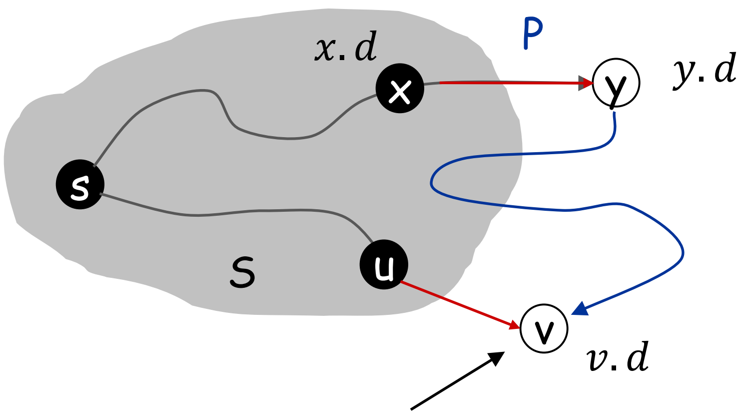
- Consider a shortest path P from s to v, shown in the right graph and mapped to the left
  - P is a shortest path, so its subpath $(s,\ldots,x,y)$ must also be a shortest path. According to the cut and paste argument, $\delta(s,y) = \delta(s,x) + w(x,y)$ --- ❹
  - Since $x \in S$, we have $x.d = \delta(s,x)$, so $\delta(s,y) = x.d + w(x,y)$--- ❺.

Dijkstra's Algorithm: Correctness

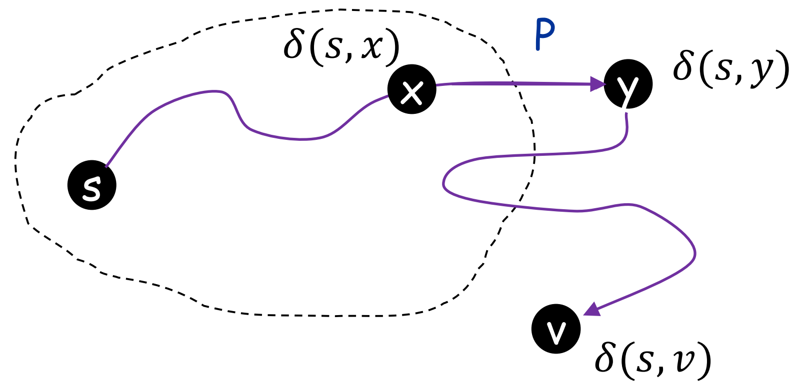
Lemma. Suppose $u.d = \delta(s,u)$ for all $u \in S$, and all edges leaving S have been relaxed. Then $v.d = \delta(s,v)$, where v is the vertex with minimum $v.d$ in $V - S$.

Pf. (by contradiction)

- Note that $v.d$ starts $= \infty$. Whenever $v.d$ is updated, it's because a path with distance $v.d$ was found. So always have $v.d \geq \delta(s,v)$.
  Thus if $v.d \neq \delta(s,v)$ then $v.d > \delta(s,v)$ --- ❶.



$v.d$ is the smallest among nodes not in S



A shortest path from s to from v

- Consider a shortest path P from s to v, shown in the right graph and mapped to the left
  - In the left graph, the edge $x \rightarrow y$ has been relaxed, so $y.d \leq x.d + w(x,y)$ --- ❻.

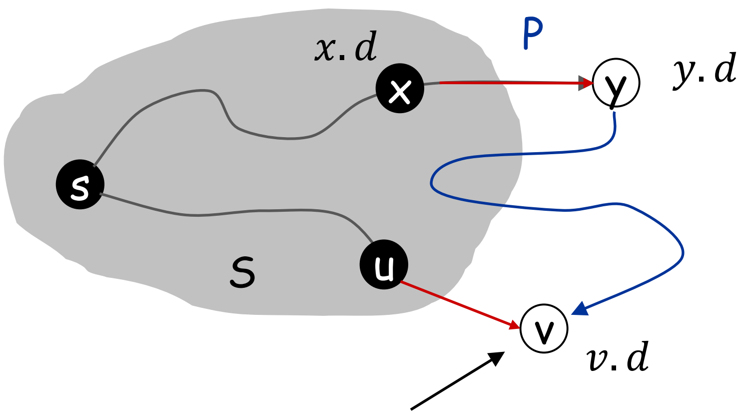
After each round of relaxation, $j.d = \min\limits_{i:(i,j)\in E} \{i.d + w(i,j)\}$, so $j.d \leq i.d + w(i,j), (i,j) \in E$

Dijkstra's Algorithm: Correctness

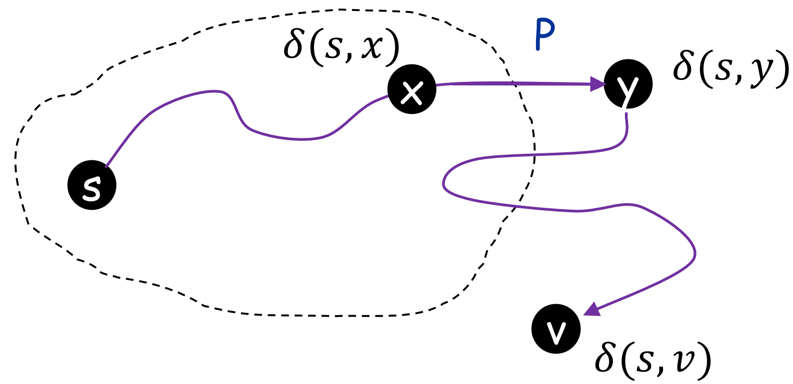
Lemma. Suppose $u.d = \delta(s,u)$ for all $u \in S$, and all edges leaving S have been relaxed. Then $v.d = \delta(s,v)$, where v is the vertex with minimum $v.d$ in $V - S$.

Pf. (by contradiction)

- Note that $v.d$ starts $= \infty$. Whenever $v.d$ is updated, it's because a path with distance $v.d$ was found. So always have $v.d \geq \delta(s,v)$.
  Thus if $v.d \neq \delta(s,v)$ then $v.d > \delta(s,v)$ --- ❶.



$v.d$ is the smallest among nodes not in S



A shortest path from s to from v

- Consider a shortest path P from s to v, shown in the right graph and mapped to the left

=> $v.d > \delta(s,v) = \delta(s,y) + \delta(y,v) \geq \delta(s,y) = \delta(s,x) + w(x,y) = x.d + w(x,y) \geq y.d$,

❶ ❷ ❸ ❹ ❺ ❻

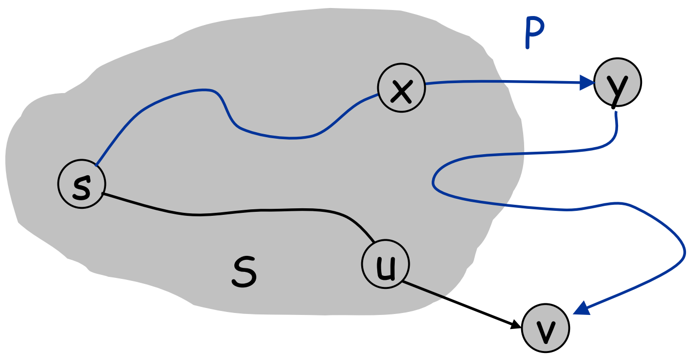
contradicting fact that $v.d$ is the **smallest** in $V - S$.

Dijkstra's Algorithm: Correctness (Original Slide)

Lemma. Suppose $u.d = \delta(s,u)$ for all $u \in S$, and all edges leaving S have been relaxed. Then $v.d = \delta(s,v)$, where v is the vertex with minimum $v.d$ in $V - S$.

Pf. (by contradiction)

- Note that $v.d$ starts $= \infty$. Whenever $v.d$ is updated, it's because a path with distance $v.d$ was found. So always have $v.d \geq \delta(s,v)$.
  Thus if $v.d \neq \delta(s,v)$ then $v.d > \delta(s,v)$.



- Consider the shortest path P from s to v.
  - Suppose $x \to y$ is the first edge on P that takes P out of S.
  - Since $x \in S$, we have $x.d = \delta(s,x)$.
  - The edge $x \to y$ has been relaxed, so $y.d \leq x.d + w(x,y)$.
  - P is shortest path, its subpath $(s, \ldots, x, y)$ must also be shortest, so
    $$x.d + w(x,y) = \delta(s,y).$$
  - $\delta(s,y) \leq \delta(s,v)$, assuming nonnegative weights

=> $v.d > \delta(s,v) \quad \geq \delta(s,y) \quad = \quad x.d + w(x,y) \quad \geq \quad y.d,$

contradicting fact that $v.d$ is the **smallest** in $V - S$.