BFS is a single-source shortest-path algorithm that works on unweighted graphs, that is, graphs in which each edge has unit weight.

Shortest Path Algorithms

?? Minimize weights ??

Time, cost, penalties, loss, etc.

Introduction

- Given a weighted, directed graph $\boldsymbol{G = (V, E)}$, with weight function $\boldsymbol{w : E \rightarrow \mathbb{R}}$.
- $\boldsymbol{w(p)}$, the weight of path $\boldsymbol{p}$ from $\boldsymbol{v_0}$ to $\boldsymbol{v_k}$ is given by

$$w(p) = \sum_{i=1}^{k} w(v_{i-1}, v_i)$$

- Then shortest-path weight $\boldsymbol{\delta(u,v)}$ is defined as

$$\delta(u, v) = \begin{cases} \min\{w(p) : u \overset{p}{\leadsto} v\} & \text{if there is a path from } u \text{ to } v \text{ ,} \\ \infty & \text{otherwise .} \end{cases}$$

- Shortest path from vertex $\boldsymbol{u}$ to vertex $\boldsymbol{v}$ is then defined as any path $\boldsymbol{p}$ with weight $\boldsymbol{w(p) = \delta(u,v)}$.

Contd…

- Single-source shortest-paths problem, i.e. given a graph find a shortest path from a given source vertex to each other vertex.
  - Dijkstra's algorithm.
- Variants:
  - Single-destination shortest-paths problem
  - Single-pair shortest-path problem
  - All-pairs shortest-paths problem, i.e. find a shortest path from u to v for every pair of vertices u and v.
    - Floyd-Warshall algorithm.

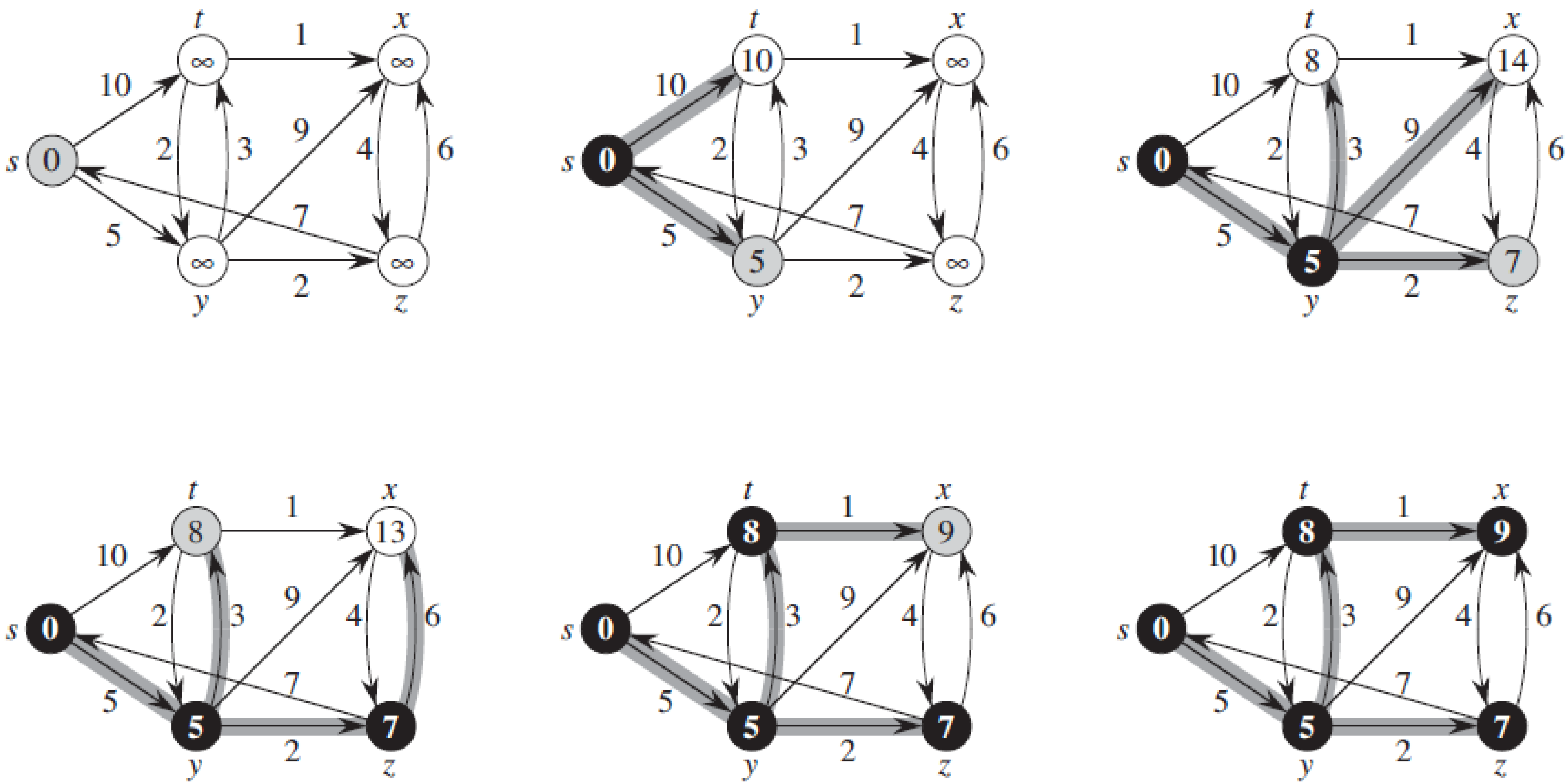
Dijkstra’s Algorithm

- Solves single-source shortest-paths problem on a weighted, directed graph in which all edge weights are nonnegative.

Example

RELAX(u, v, w)

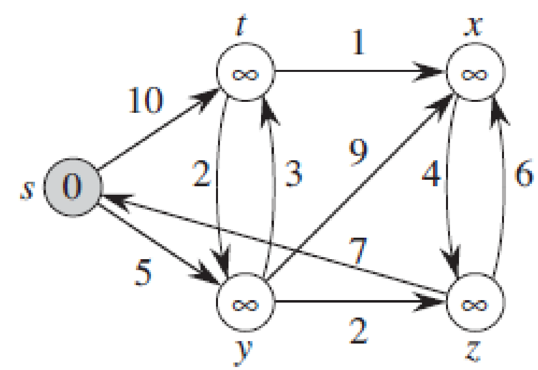
if $v.d > u.d + w(u, v)$
  $v.d = u.d + w(u, v)$
  $v.\pi = u$



Implementation

s → t | 10 → y | 5
t → x | 1 → y | 2
x → z | 4
y → t | 3 → x | 9 → z | 2
z → s | 7 → x | 6

```
DIJKSTRA(G, w, s)
INITIALIZE-SINGLE-SOURCE(G, s)
S = ∅
Q = G.V
while Q ≠ ∅
    u = EXTRACT-MIN(Q)
    S = S ∪ {u}
    for each vertex v ∈ G.Adj[u]
        RELAX(u, v, w)
```




```
INITIALIZE-SINGLE-SOURCE(G, s)
for each vertex v ∈ G.V
    v.d = ∞
    v.π = NIL
s.d = 0
```

```
RELAX(u, v, w)
if v.d > u.d + w(u, v)
    v.d = u.d + w(u, v)
    v.π = u
```

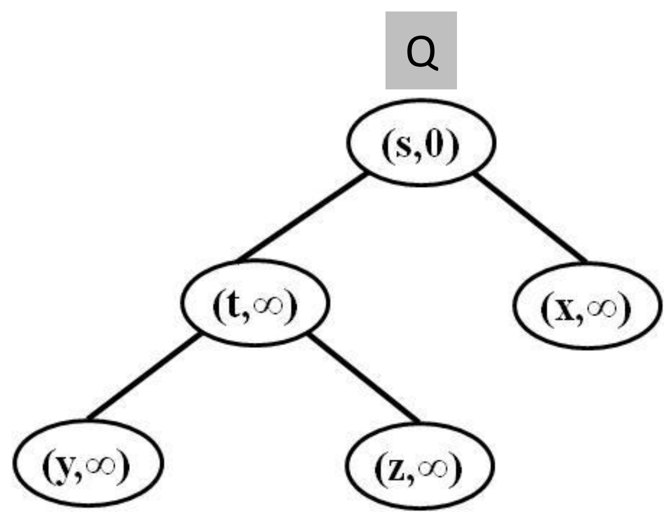
Example - Execution

s → t 10 → y 5
t → x 1 → y 2
x → z 4
y → t 3 → x 9 → z 2
z → s 7 → x 6

S = {}

Vertex	π	d
s	NIL	0
t	NIL	∞
x	NIL	∞
y	NIL	∞
z	NIL	∞

```
DIJKSTRA(G, w, s)
INITIALIZE-SINGLE-SOURCE(G, s)
S = ∅
Q = G.V
while Q ≠ ∅
    u = EXTRACT-MIN(Q)
    S = S ∪ {u}
    for each vertex v ∈ G.Adj[u]
        RELAX(u, v, w)
```



Example - Execution

S = {s}

Vertex	π	d
s	NIL	0
t	NIL	∞
x	NIL	∞
y	NIL	∞
z	NIL	∞

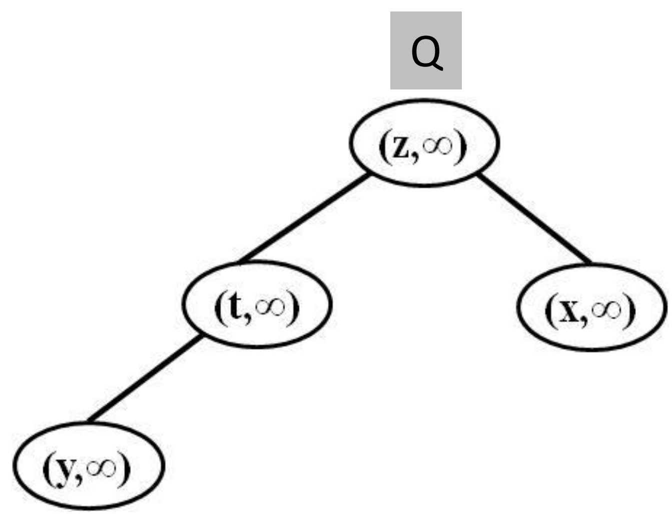
s → t 10 → y 5
t → x 1 → y 2
x → z 4
y → t 3 → x 9 → z 2
z → s 7 → x 6

RELAX(u, v, w)

```
if v.d > u.d + w(u,v)
    v.d = u.d + w(u,v)
    v.π = u
```

DIJKSTRA(G, w, s)

```
INITIALIZE-SINGLE-SOURCE(G, s)
S = ∅
Q = G.V
while Q ≠ ∅
    u = EXTRACT-MIN(Q)
    S = S ∪ {u}
    for each vertex v ∈ G.Adj[u]
        RELAX(u, v, w)
```



Example - Execution

S = {s}

Vertex	π	d
s	NIL	0
t	s	10
x	NIL	∞
y	NIL	∞
z	NIL	∞

s → [t | 10] → [y | 5]
t → [x | 1] → [y | 2]
x → [z | 4]
y → [t | 3] → [x | 9] → [z | 2]
z → [s | 7] → [x | 6]

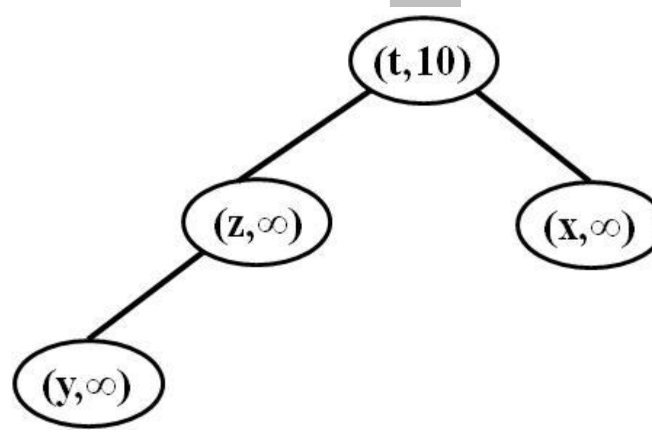
RELAX(u, v, w)

```
if v.d > u.d + w(u, v)
    v.d = u.d + w(u, v)
    v.π = u
```

DIJKSTRA(G, w, s)

```
INITIALIZE-SINGLE-SOURCE(G, s)
S = ∅
Q = G.V
while Q ≠ ∅
    u = EXTRACT-MIN(Q)
    S = S ∪ {u}
    for each vertex v ∈ G.Adj[u]
        RELAX(u, v, w)
```

Q



Example - Execution

S = {s}

Vertex	π	d
s	NIL	0
t	s	10
x	NIL	∞
y	s	5
z	NIL	∞

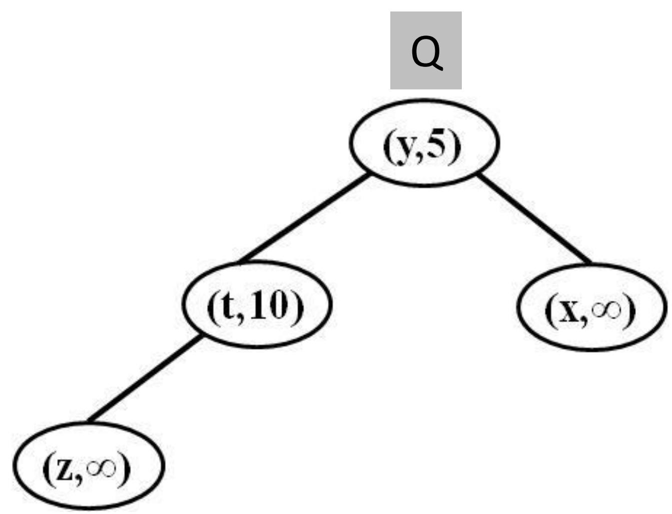
s → t 10 → y 5
t → x 1 → y 2
x → z 4
y → t 3 → x 9 → z 2
z → s 7 → x 6

RELAX(u, v, w)

```
if v.d > u.d + w(u, v)
    v.d = u.d + w(u, v)
    v.π = u
```

DIJKSTRA(G, w, s)

```
INITIALIZE-SINGLE-SOURCE(G, s)
S = ∅
Q = G.V
while Q ≠ ∅
    u = EXTRACT-MIN(Q)
    S = S ∪ {u}
    for each vertex v ∈ G.Adj[u]
        RELAX(u, v, w)
```



Example - Execution

s → t | 10 → y | 5
t → x | 1 → y | 2
x → z | 4
y → t | 3 → x | 9 → z | 2
z → s | 7 → x | 6

S = {s, y}

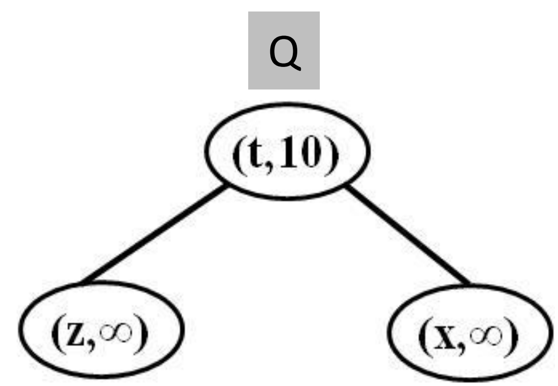
Vertex	π	d
s	NIL	0
t	s	10
x	NIL	∞
y	s	5
z	NIL	∞

RELAX(u, v, w)

```
if v.d > u.d + w(u, v)
    v.d = u.d + w(u, v)
    v.π = u
```

DIJKSTRA(G, w, s)

```
INITIALIZE-SINGLE-SOURCE(G, s)
S = ∅
Q = G.V
while Q ≠ ∅
    u = EXTRACT-MIN(Q)
    S = S ∪ {u}
    for each vertex v ∈ G.Adj[u]
        RELAX(u, v, w)
```



Example - Execution

s → (t, 10) → (y, 5)
t → (x, 1) → (y, 2)
x → (z, 4)
y → (t, 3) → (x, 9) → (z, 2)
z → (s, 7) → (x, 6)

S = {s, y}

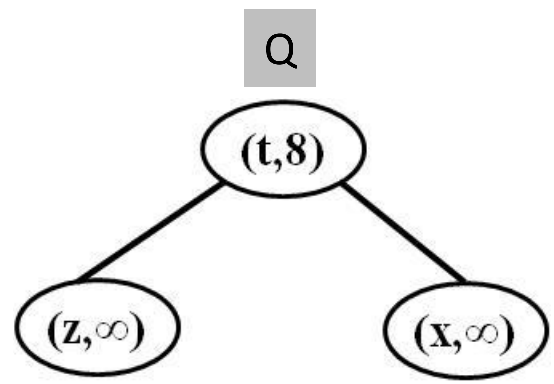
Vertex	π	d
s	NIL	0
t	y	8
x	NIL	∞
y	s	5
z	NIL	∞

RELAX(u, v, w)

```
if v.d > u.d + w(u, v)
    v.d = u.d + w(u, v)
    v.π = u
```

DIJKSTRA(G, w, s)

```
INITIALIZE-SINGLE-SOURCE(G, s)
S = ∅
Q = G.V
while Q ≠ ∅
    u = EXTRACT-MIN(Q)
    S = S ∪ {u}
    for each vertex v ∈ G.Adj[u]
        RELAX(u, v, w)
```



Example - Execution

s	→	t	10	→	y	5			
t	→	x	1	→	y	2			
x	→	z	4						
y	→	t	3	→	x	9	→	z	2
z	→	s	7	→	x	6			

S = {s, y}

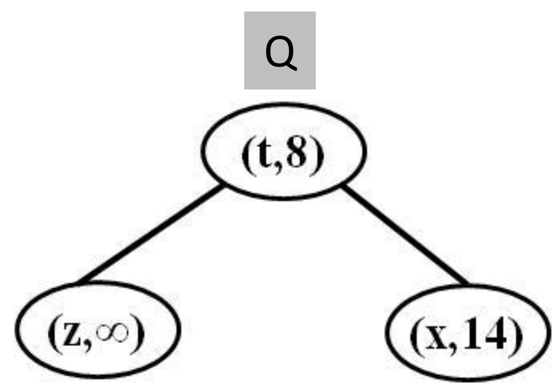
Vertex	π	d
s	NIL	0
t	y	8
x	y	14
y	s	5
z	NIL	∞

RELAX(u, v, w)

```
if v.d > u.d + w(u, v)
    v.d = u.d + w(u, v)
    v.π = u
```

DIJKSTRA(G, w, s)

```
INITIALIZE-SINGLE-SOURCE(G, s)
S = ∅
Q = G.V
while Q ≠ ∅
    u = EXTRACT-MIN(Q)
    S = S ∪ {u}
    for each vertex v ∈ G.Adj[u]
        RELAX(u, v, w)
```



Example - Execution

s → t 10 → y 5
t → x 1 → y 2
x → z 4
y → t 3 → x 9 → z 2
z → s 7 → x 6

S = {s, y}

Vertex	π	d
s	NIL	0
t	y	8
x	y	14
y	s	5
z	y	7

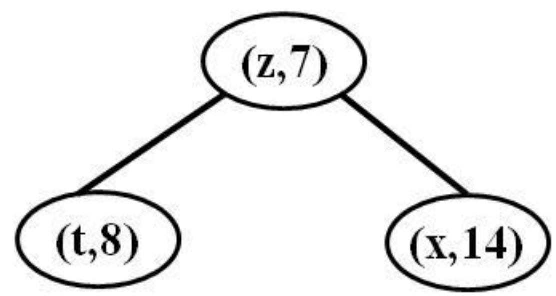
RELAX(u, v, w)

```
if v.d > u.d + w(u, v)
    v.d = u.d + w(u, v)
    v.π = u
```

DIJKSTRA(G, w, s)

```
INITIALIZE-SINGLE-SOURCE(G, s)
S = ∅
Q = G.V
while Q ≠ ∅
    u = EXTRACT-MIN(Q)
    S = S ∪ {u}
    for each vertex v ∈ G.Adj[u]
        RELAX(u, v, w)
```

Q



Example - Execution

s → t 10 → y 5
t → x 1 → y 2
x → z 4
y → t 3 → x 9 → z 2
z → s 7 → x 6

S = {s, y, z}

Vertex	π	d
s	NIL	0
t	y	8
x	y	14
y	s	5
z	y	7

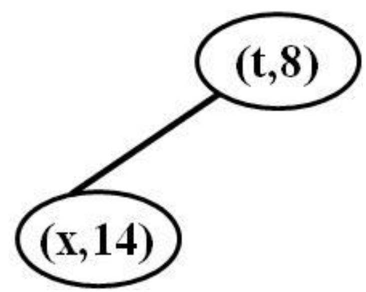
RELAX(u, v, w)

```
if v.d > u.d + w(u, v)
    v.d = u.d + w(u, v)
    v.π = u
```

DIJKSTRA(G, w, s)

```
INITIALIZE-SINGLE-SOURCE(G, s)
S = ∅
Q = G.V
while Q ≠ ∅
    u = EXTRACT-MIN(Q)
    S = S ∪ {u}
    for each vertex v ∈ G.Adj[u]
        RELAX(u, v, w)
```

Q



Example - Execution

s → t 10 → y 5
t → x 1 → y 2
x → z 4
y → t 3 → x 9 → z 2
z → s 7 → x 6

S = {s, y, z}

Vertex	π	d
s	NIL	0
t	y	8
x	z	13
y	s	5
z	y	7

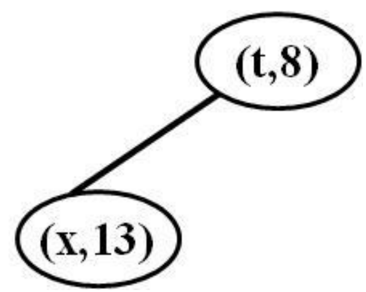
RELAX(u, v, w)

```
if v.d > u.d + w(u, v)
    v.d = u.d + w(u, v)
    v.π = u
```

DIJKSTRA(G, w, s)

```
INITIALIZE-SINGLE-SOURCE(G, s)
S = ∅
Q = G.V
while Q ≠ ∅
    u = EXTRACT-MIN(Q)
    S = S ∪ {u}
    for each vertex v ∈ G.Adj[u]
        RELAX(u, v, w)
```

Q



Example - Execution

S = {s, y, z, t}

Vertex	π	d
s	NIL	0
t	y	8
x	z	13
y	s	5
z	y	7

s → | t | 10 | → | y | 5 |

t → | x | 1 | → | y | 2 |

x → | z | 4 |

y → | t | 3 | → | x | 9 | → | z | 2 |

z → | s | 7 | → | x | 6 |

RELAX(u, v, w)

```
if v.d > u.d + w(u,v)
    v.d = u.d + w(u,v)
    v.π = u
```

Q

(x,13)

DIJKSTRA(G, w, s)

```
INITIALIZE-SINGLE-SOURCE(G, s)
S = ∅
Q = G.V
while Q ≠ ∅
    u = EXTRACT-MIN(Q)
    S = S ∪ {u}
    for each vertex v ∈ G.Adj[u]
        RELAX(u, v, w)
```

Example - Execution

$S = \{s, y, z, t\}$

Vertex	π	d
s	NIL	0
t	y	8
x	t	9
y	s	5
z	y	7

s → [t | 10] → [y | 5]
t → [x | 1] → [y | 2]
x → [z | 4]
y → [t | 3] → [x | 9] → [z | 2]
z → [s | 7] → [x | 6]

RELAX(u, v, w)

```
if v.d > u.d + w(u, v)
    v.d = u.d + w(u, v)
    v.π = u
```

DIJKSTRA(G, w, s)

```
INITIALIZE-SINGLE-SOURCE(G, s)
S = ∅
Q = G.V
while Q ≠ ∅
    u = EXTRACT-MIN(Q)
    S = S ∪ {u}
    for each vertex v ∈ G.Adj[u]
        RELAX(u, v, w)
```

Q

(x,9)

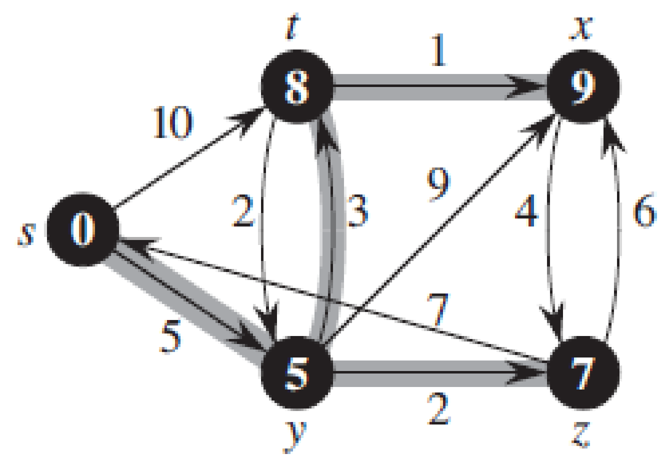
Example - Execution

s → t 10 → y 5
t → x 1 → y 2
x → z 4
y → t 3 → x 9 → z 2
z → s 7 → x 6

S = {s, y, z, t, x}

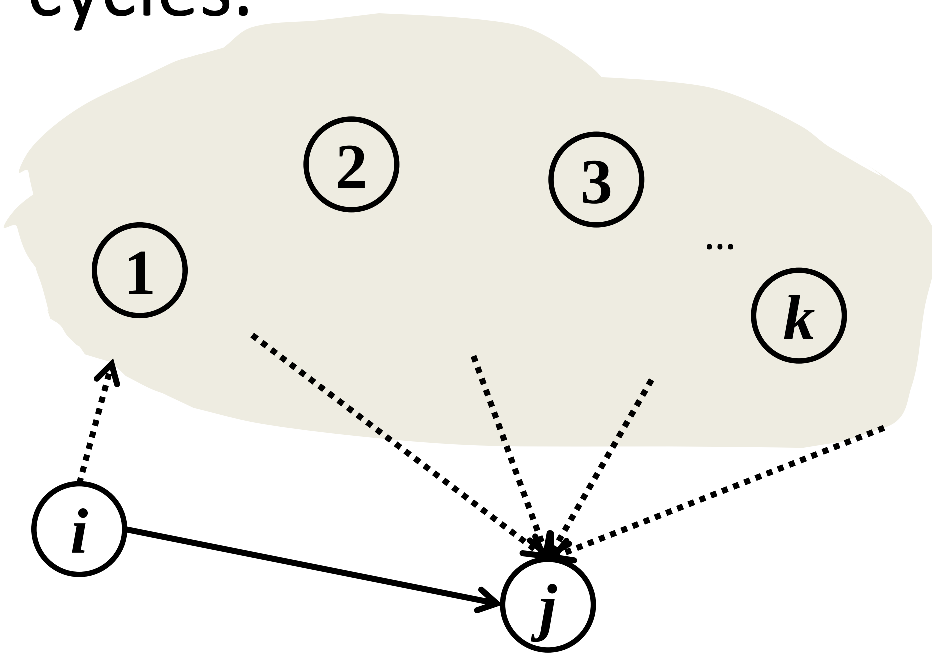
Vertex	π	d
s	NIL	0
t	y	8
x	t	9
y	s	5
z	y	7

```
DIJKSTRA(G, w, s)
INITIALIZE-SINGLE-SOURCE(G, s)
S = ∅
Q = G.V
while Q ≠ ∅
    u = EXTRACT-MIN(Q)
    S = S ∪ {u}
    for each vertex v ∈ G.Adj[u]
        RELAX(u, v, w)
```

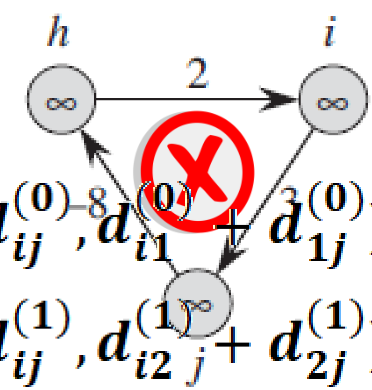


Floyd-Warshall Algorithm

- It a dynamic-programming formulation to solve the all-pairs shortest-paths problem on a directed graph, which may have negative-weight edges, but it is assumed that there are no negative-weight cycles.



$$d_{ij}^{(0)} = w_{ij}$$



$$d_{ij}^{(1)} = min(d_{ij}^{(0)}, d_{i1}^{(0)} + d_{1j}^{(0)})$$

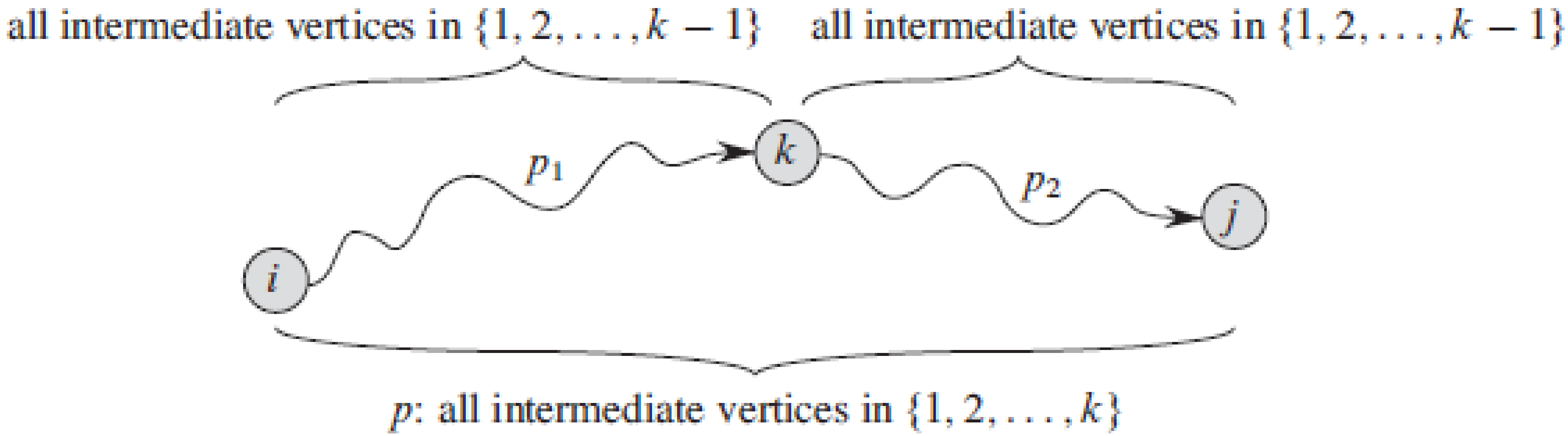
$$d_{ij}^{(2)} = min(d_{ij}^{(1)}, d_{i2}^{(1)} + d_{2j}^{(1)})$$

$$d_{ij}^{(3)} = min(d_{ij}^{(2)}, d_{i3}^{(2)} + d_{3j}^{(2)})$$

...

$$d_{ij}^{(k)} = min(d_{ij}^{(k-1)}, d_{ik}^{(k-1)} + d_{kj}^{(k-1)})$$

Contd…



$$d_{ij}^{(k)} = min(d_{ij}^{(k-1)}, d_{ik}^{(k-1)} + d_{kj}^{(k-1)})$$

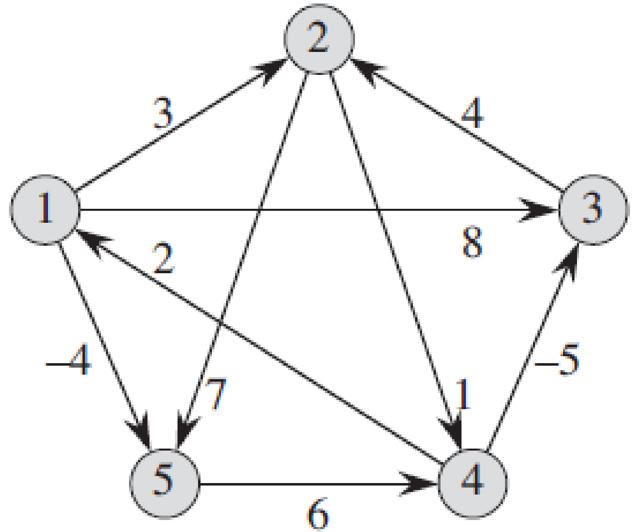
Contd…

$$d_{ij}^{(k)} = \begin{cases} w_{ij} & \text{if } k = 0\,, \\ \min\left(d_{ij}^{(k-1)}, d_{ik}^{(k-1)} + d_{kj}^{(k-1)}\right) & \text{if } k \geq 1\,. \end{cases}$$

$$\pi_{ij}^{(0)} = \begin{cases} \text{NIL} & \text{if } i = j \text{ or } w_{ij} = \infty\,, \\ i & \text{if } i \neq j \text{ and } w_{ij} < \infty\,. \end{cases}$$

$$\pi_{ij}^{(k)} = \begin{cases} \pi_{ij}^{(k-1)} & \text{if } d_{ij}^{(k-1)} \leq d_{ik}^{(k-1)} + d_{kj}^{(k-1)}\,, \\ \pi_{kj}^{(k-1)} & \text{if } d_{ij}^{(k-1)} > d_{ik}^{(k-1)} + d_{kj}^{(k-1)}\,. \end{cases}$$

Example



$$d_{ij}^{(k)} = \begin{cases} w_{ij} & \text{if } k = 0 , \\ \min\left(d_{ij}^{(k-1)}, d_{ik}^{(k-1)} + d_{kj}^{(k-1)}\right) & \text{if } k \geq 1 . \end{cases}$$

$$\pi_{ij}^{(0)} = \begin{cases} \text{NIL} & \text{if } i = j \text{ or } w_{ij} = \infty , \\ i & \text{if } i \neq j \text{ and } w_{ij} < \infty . \end{cases}$$

$D^{(0)}$	V_1	V_2	V_3	V_4	V_5
V_1	0	3	8	∞	-4
V_2	∞	0	∞	1	7
V_3	∞	4	0	∞	∞
V_4	2	∞	-5	0	∞
V_5	∞	∞	∞	6	0

$\pi^{(0)}$	V_1	V_2	V_3	V_4	V_5
V_1	NIL	1	1	NIL	1
V_2	NIL	NIL	NIL	2	2
V_3	NIL	3	NIL	NIL	NIL
V_4	4	NIL	4	NIL	NIL
V_5	NIL	NIL	NIL	5	NIL

Contd...

$$d_{ij}^{(k)} = \begin{cases} w_{ij} & \text{if } k = 0, \\ \min\left(d_{ij}^{(k-1)}, d_{ik}^{(k-1)} + d_{kj}^{(k-1)}\right) & \text{if } k \geq 1. \end{cases}$$

$$\pi_{ij}^{(k)} = \begin{cases} \pi_{ij}^{(k-1)} & \text{if } d_{ij}^{(k-1)} \leq d_{ik}^{(k-1)} + d_{kj}^{(k-1)}, \\ \pi_{kj}^{(k-1)} & \text{if } d_{ij}^{(k-1)} > d_{ik}^{(k-1)} + d_{kj}^{(k-1)}. \end{cases}$$

$D^{(0)}$	V_1	V_2	V_3	V_4	V_5
V_1	0	3	8	∞	-4
V_2	∞	0	∞	1	7
V_3	∞	4	0	∞	∞
V_4	2	∞	-5	0	∞
V_5	∞	∞	∞	6	0

$\pi^{(0)}$	V_1	V_2	V_3	V_4	V_5
V_1	NIL	1	1	NIL	1
V_2	NIL	NIL	NIL	2	2
V_3	NIL	3	NIL	NIL	NIL
V_4	4	NIL	4	NIL	NIL
V_5	NIL	NIL	NIL	5	NIL

$D^{(1)}$	V_1	V_2	V_3	V_4	V_5
V_1	0	3	8	∞	-4
V_2	∞	0	∞	1	7
V_3	∞	4	0	∞	∞
V_4	2	5	-5	0	-2
V_5	∞	∞	∞	6	0

$\pi^{(1)}$	V_1	V_2	V_3	V_4	V_5
V_1	NIL	1	1	NIL	1
V_2	NIL	NIL	NIL	2	2
V_3	NIL	3	NIL	NIL	NIL
V_4	4	1	4	NIL	1
V_5	NIL	NIL	NIL	5	NIL

Contd...

$$d_{ij}^{(k)} = \begin{cases} w_{ij} & \text{if } k = 0, \\ \min\left(d_{ij}^{(k-1)}, d_{ik}^{(k-1)} + d_{kj}^{(k-1)}\right) & \text{if } k \geq 1. \end{cases}$$

$$\pi_{ij}^{(k)} = \begin{cases} \pi_{ij}^{(k-1)} & \text{if } d_{ij}^{(k-1)} \leq d_{ik}^{(k-1)} + d_{kj}^{(k-1)}, \\ \pi_{kj}^{(k-1)} & \text{if } d_{ij}^{(k-1)} > d_{ik}^{(k-1)} + d_{kj}^{(k-1)}. \end{cases}$$

$D^{(1)}$	V_1	V_2	V_3	V_4	V_5
V_1	0	3	8	∞	-4
V_2	∞	0	∞	1	7
V_3	∞	4	0	∞	∞
V_4	2	5	-5	0	-2
V_5	∞	∞	∞	6	0

$\pi^{(1)}$	V_1	V_2	V_3	V_4	V_5
V_1	NIL	1	1	NIL	1
V_2	NIL	NIL	NIL	2	2
V_3	NIL	3	NIL	NIL	NIL
V_4	4	1	4	NIL	1
V_5	NIL	NIL	NIL	5	NIL

$D^{(2)}$	V_1	V_2	V_3	V_4	V_5
V_1	0	3	8	4	-4
V_2	∞	0	∞	1	7
V_3	∞	4	0	5	11
V_4	2	5	-5	0	-2
V_5	∞	∞	∞	6	0

$\pi^{(2)}$	V_1	V_2	V_3	V_4	V_5
V_1	NIL	1	1	2	1
V_2	NIL	NIL	NIL	2	2
V_3	NIL	3	NIL	2	2
V_4	4	1	4	NIL	1
V_5	NIL	NIL	NIL	5	NIL

Contd…

$$d_{ij}^{(k)} = \begin{cases} w_{ij} & \text{if } k = 0\,, \\ \min\left(d_{ij}^{(k-1)}, d_{ik}^{(k-1)} + d_{kj}^{(k-1)}\right) & \text{if } k \ge 1\,. \end{cases}$$

$$\pi_{ij}^{(k)} = \begin{cases} \pi_{ij}^{(k-1)} & \text{if } d_{ij}^{(k-1)} \le d_{ik}^{(k-1)} + d_{kj}^{(k-1)}\,, \\ \pi_{kj}^{(k-1)} & \text{if } d_{ij}^{(k-1)} > d_{ik}^{(k-1)} + d_{kj}^{(k-1)}\,. \end{cases}$$

$D^{(2)}$	V_1	V_2	V_3	V_4	V_5
V_1	0	3	8	4	-4
V_2	∞	0	∞	1	7
V_3	∞	4	0	5	11
V_4	2	5	-5	0	-2
V_5	∞	∞	∞	6	0

$\pi^{(2)}$	V_1	V_2	V_3	V_4	V_5
V_1	NIL	1	1	2	1
V_2	NIL	NIL	NIL	2	2
V_3	NIL	3	NIL	2	2
V_4	4	1	4	NIL	1
V_5	NIL	NIL	NIL	5	NIL

$D^{(3)}$	V_1	V_2	V_3	V_4	V_5
V_1	0	3	8	4	-4
V_2	∞	0	∞	1	7
V_3	∞	4	0	5	11
V_4	2	-1	-5	0	-2
V_5	∞	∞	∞	6	0

$\pi^{(3)}$	V_1	V_2	V_3	V_4	V_5
V_1	NIL	1	1	2	1
V_2	NIL	NIL	NIL	2	2
V_3	NIL	3	NIL	2	2
V_4	4	3	4	NIL	1
V_5	NIL	NIL	NIL	5	NIL

Contd...

$$d_{ij}^{(k)} = \begin{cases} w_{ij} & \text{if } k = 0\,, \\ \min\left(d_{ij}^{(k-1)}, d_{ik}^{(k-1)} + d_{kj}^{(k-1)}\right) & \text{if } k \geq 1\,. \end{cases} \qquad \pi_{ij}^{(k)} = \begin{cases} \pi_{ij}^{(k-1)} & \text{if } d_{ij}^{(k-1)} \leq d_{ik}^{(k-1)} + d_{kj}^{(k-1)}\,, \\ \pi_{kj}^{(k-1)} & \text{if } d_{ij}^{(k-1)} > d_{ik}^{(k-1)} + d_{kj}^{(k-1)}\,. \end{cases}$$

$D^{(3)}$	V_1	V_2	V_3	V_4	V_5
V_1	0	3	8	4	-4
V_2	∞	0	∞	1	7
V_3	∞	4	0	5	11
V_4	2	-1	-5	0	-2
V_5	∞	∞	∞	6	0

$\pi^{(3)}$	V_1	V_2	V_3	V_4	V_5
V_1	NIL	1	1	2	1
V_2	NIL	NIL	NIL	2	2
V_3	NIL	3	NIL	2	2
V_4	4	3	4	NIL	1
V_5	NIL	NIL	NIL	5	NIL

$D^{(4)}$	V_1	V_2	V_3	V_4	V_5
V_1	0	3	-1	4	-4
V_2	3	0	-4	1	-1
V_3	7	4	0	5	3
V_4	2	-1	-5	0	-2
V_5	8	5	1	6	0

$\pi^{(4)}$	V_1	V_2	V_3	V_4	V_5
V_1	NIL	1	4	2	1
V_2	4	NIL	4	2	1
V_3	4	3	NIL	2	1
V_4	4	3	4	NIL	1
V_5	4	3	4	5	NIL

Contd...

$$d_{ij}^{(k)} = \begin{cases} w_{ij} & \text{if } k = 0\,, \\ \min\left(d_{ij}^{(k-1)}, d_{ik}^{(k-1)} + d_{kj}^{(k-1)}\right) & \text{if } k \ge 1\,. \end{cases} \qquad \pi_{ij}^{(k)} = \begin{cases} \pi_{ij}^{(k-1)} & \text{if } d_{ij}^{(k-1)} \le d_{ik}^{(k-1)} + d_{kj}^{(k-1)}\,, \\ \pi_{kj}^{(k-1)} & \text{if } d_{ij}^{(k-1)} > d_{ik}^{(k-1)} + d_{kj}^{(k-1)}\,. \end{cases}$$

$D^{(4)}$	V_1	V_2	V_3	V_4	V_5
V_1	0	3	-1	4	-4
V_2	3	0	-4	1	-1
V_3	7	4	0	5	3
V_4	2	-1	-5	0	-2
V_5	8	5	1	6	0

$\pi^{(4)}$	V_1	V_2	V_3	V_4	V_5
V_1	NIL	1	4	2	1
V_2	4	NIL	4	2	1
V_3	4	3	NIL	2	1
V_4	4	3	4	NIL	1
V_5	4	3	4	5	NIL

$D^{(5)}$	V_1	V_2	V_3	V_4	V_5
V_1	0	1	-3	2	-4
V_2	3	0	-4	1	-1
V_3	7	4	0	5	3
V_4	2	-1	-5	0	-2
V_5	8	5	1	6	0

$\pi^{(5)}$	V_1	V_2	V_3	V_4	V_5
V_1	NIL	3	4	5	1
V_2	4	NIL	4	2	1
V_3	4	3	NIL	2	1
V_4	4	3	4	NIL	1
V_5	4	3	4	5	NIL

Implementation

$$d_{ij}^{(k)} = \begin{cases} w_{ij} & \text{if } k = 0\,, \\ \min\left(d_{ij}^{(k-1)}, d_{ik}^{(k-1)} + d_{kj}^{(k-1)}\right) & \text{if } k \ge 1\,. \end{cases}$$

$$\pi_{ij}^{(0)} = \begin{cases} \text{NIL} & \text{if } i = j \text{ or } w_{ij} = \infty\,, \\ i & \text{if } i \ne j \text{ and } w_{ij} < \infty\,. \end{cases}$$

$$\pi_{ij}^{(k)} = \begin{cases} \pi_{ij}^{(k-1)} & \text{if } d_{ij}^{(k-1)} \le d_{ik}^{(k-1)} + d_{kj}^{(k-1)}\,, \\ \pi_{kj}^{(k-1)} & \text{if } d_{ij}^{(k-1)} > d_{ik}^{(k-1)} + d_{kj}^{(k-1)}\,. \end{cases}$$

n= W.rows
$D^0 = W$
π^0 is a matrix with nil in every entry
for i=1 to n **do**
    for j = 1 to n **do**
        if $i \neq j$ and $D^0_{i,j} < \infty$ **then**
            $\pi^0_{i,j} = i$
        end if
    end for
end for
for k=1 to n **do**
    let D^k be a new $n \times n$ matrix.
    let π^k be a new $n \times n$ matrix
    for i=1 to n **do**
        for j = 1 to n **do**
            if $d^{k-1}_{ij} \le d^{k-1}_{i,k} + d^{k-1}_{k,j}$ **then**
                $d^k_{i,j} = d^{k-1}_{i,j}$
                $\pi^k_{i,j} = \pi^{k-1}_{i,j}$
            else
                $d^k_{i,j} = d^{k-1}_{i,k} + d^{k-1}_{k,j}$
                $\pi^k_{i,j} = \pi^{k-1}_{k,j}$
            end if
        end for
    end for
end for