More on Dynamic Programming

Murat Osmanoglu

SSSP

- given a weighted graph G=(V,E) and a source vertex s in V, find the shortest path from s to every other vertex in V

- the weight of each edge fixed as 1

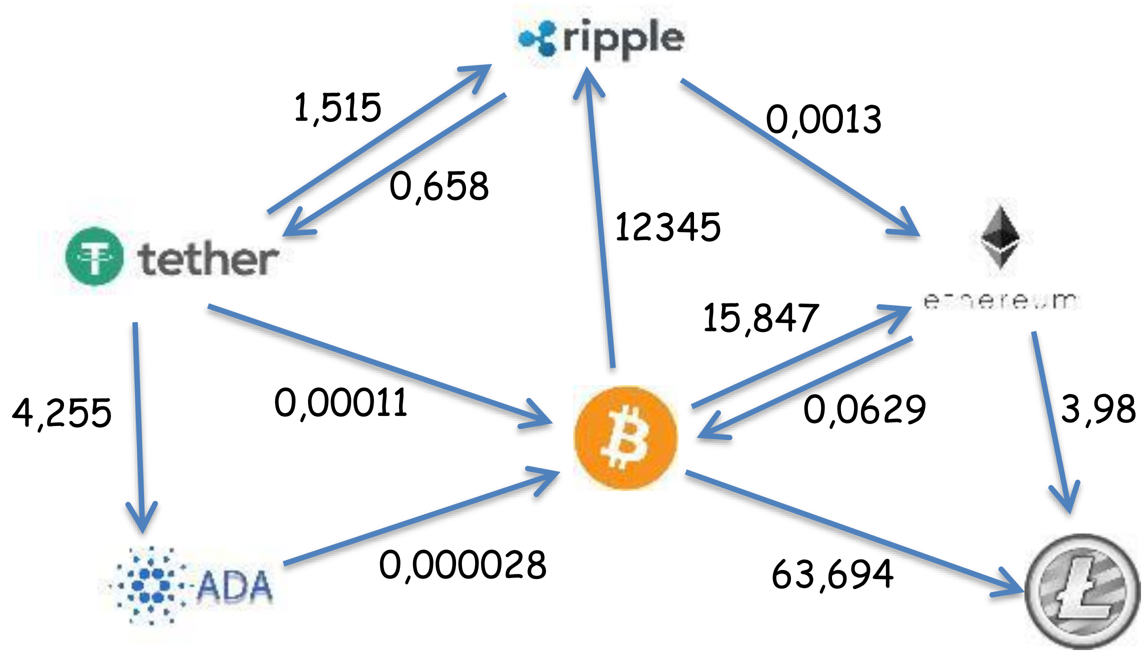
--BFS--

- the weight of each edge non-negative

--Dijkstra—

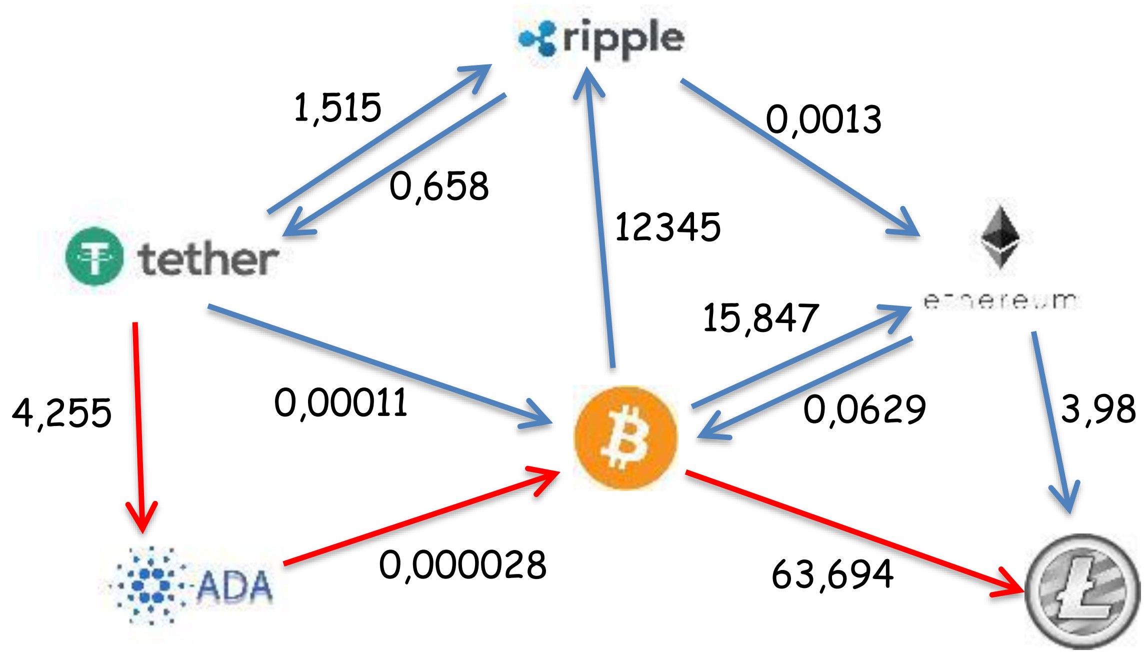
- the weight of each edge can be negative

Dijkstra's Algorithm



- find the best paths from tether to all other cryptocurrencies

Dijkstra's Algorithm

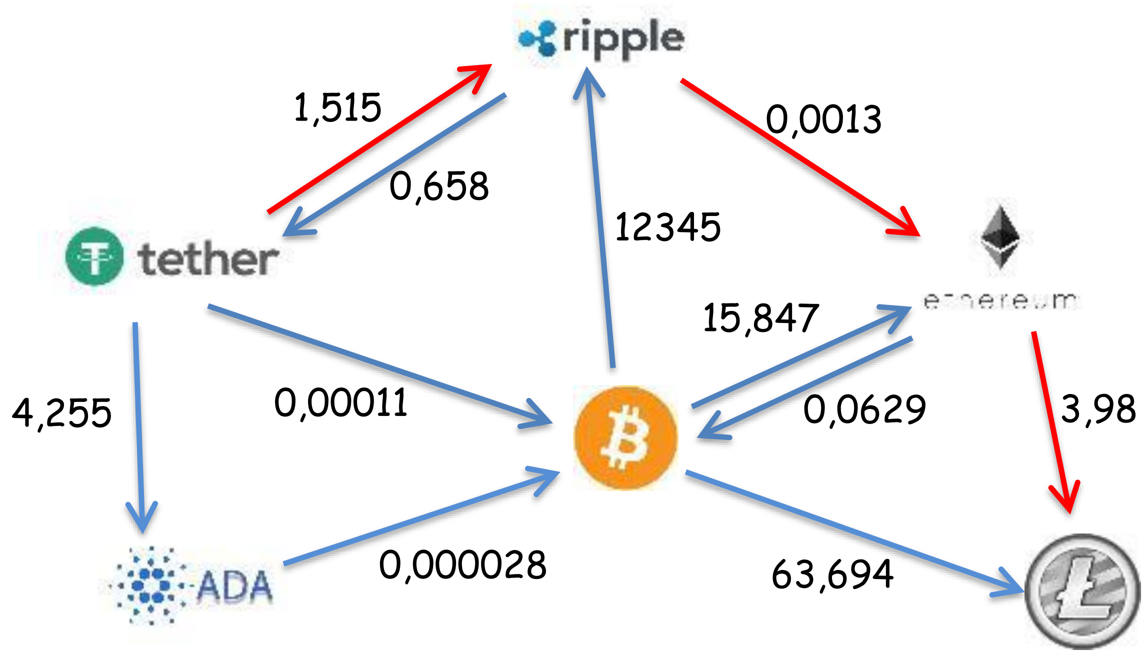


- find the best paths from tether to all other cryptocurrencies

tether – cardano – bitcoin - litecoin

1 tether = 4,255 * 0,000028 * 63,694 = 0,0075 LTC

Dijkstra's Algorithm



- find the best paths from tether to all other cryptocurrencies

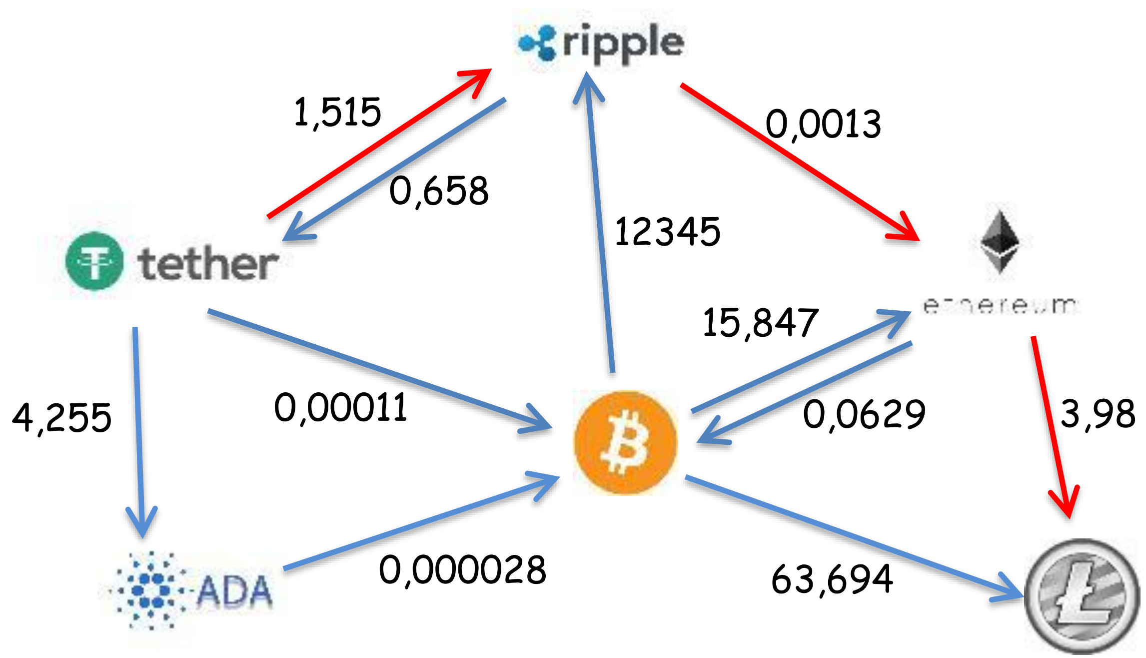
tether – cardano – bitcoin - litecoin

1 tether = 4,255 * 0,000028 * 63,694 = 0,0075 LTC

tether – ripple – ethereum - litecoin

1 tether = 1,515 * 0,0013 * 3,98 = 0,0078 LTC

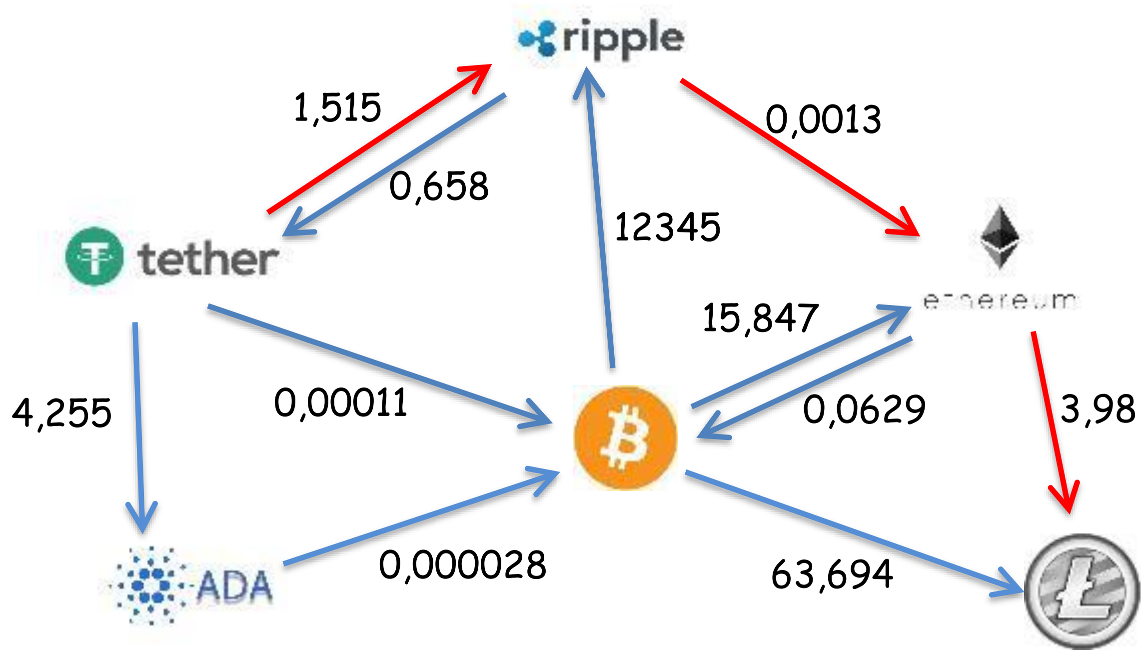
Dijkstra's Algorithm



- find the best paths from tether to all other cryptocurrencies

$$max_{P \in \{ s \to u \}} \left(\prod_{e \in P} w(e)\right)$$

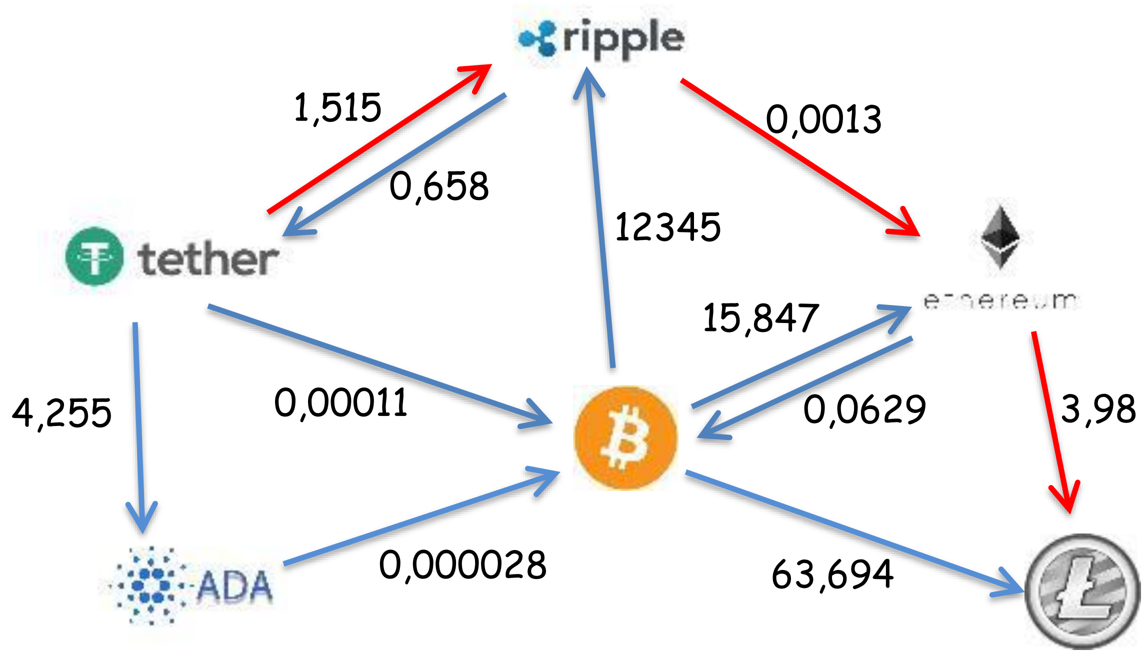
Dijkstra's Algorithm



- find the best paths from tether to all other cryptocurrencies

$$K = max_{P\,\in\{\,s\,\rightarrow u\,\}}\left(\prod_{e\,\in P} w(e)\right)$$

Dijkstra's Algorithm

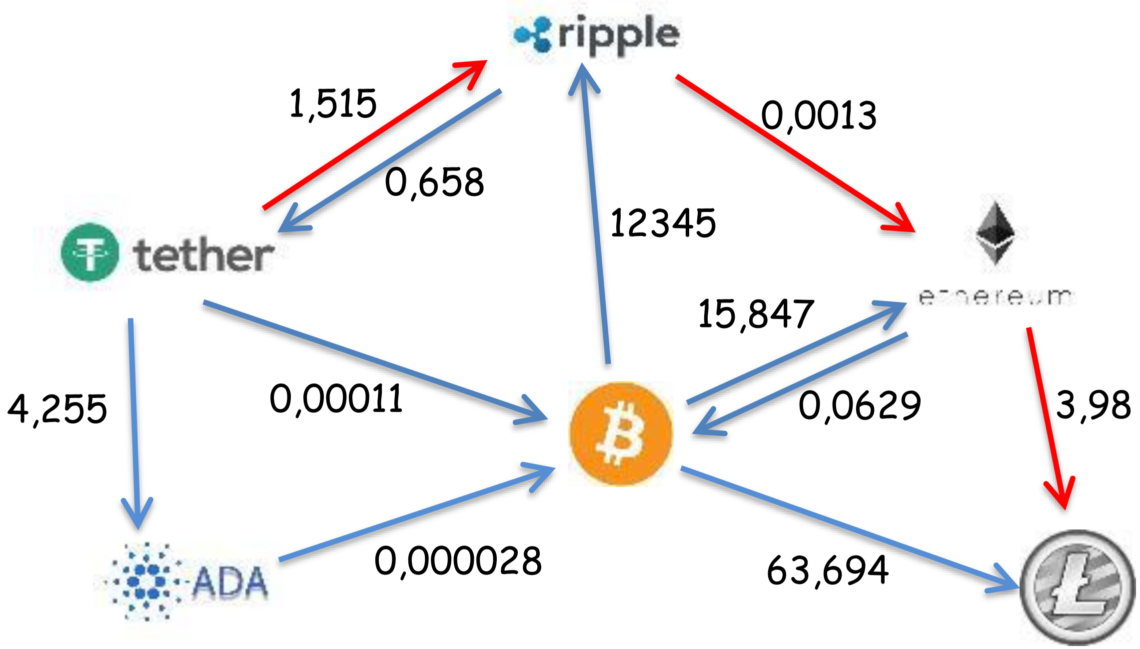


- find the best paths from tether to all other cryptocurrencies

K = $max_{P \in \{ s \to u \}} (\prod_{e \in P} w(e))$

log K = log $(max_{P \in \{ s \to u \}} \sum_{e \in P} w(e))$

Dijkstra's Algorithm



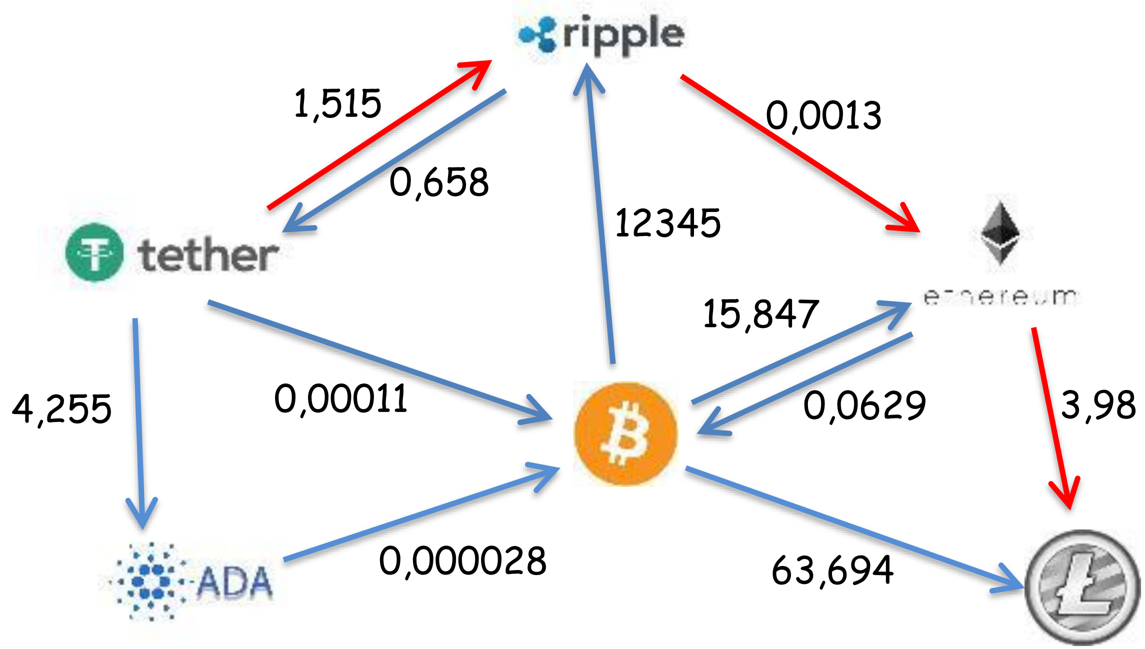
- find the best paths from tether to all other cryptocurrencies

K = $max_{P \in \{s \rightarrow u\}} (\prod_{e \in P} w(e))$

log K = log $(max_{P \in \{s \rightarrow u\}} \sum_{e \in P} w(e))$

log K = $max_{P \in \{s \rightarrow u\}}$ log $(\sum_{e \in P} w(e))$

Dijkstra's Algorithm



- find the best paths from tether to all other cryptocurrencies

K = $max_{P \in \{ s \rightarrow u \}} \left(\prod_{e \in P} w(e) \right)$

log K = log $\left(max_{P \in \{ s \rightarrow u \}} \sum_{e \in P} w(e) \right)$

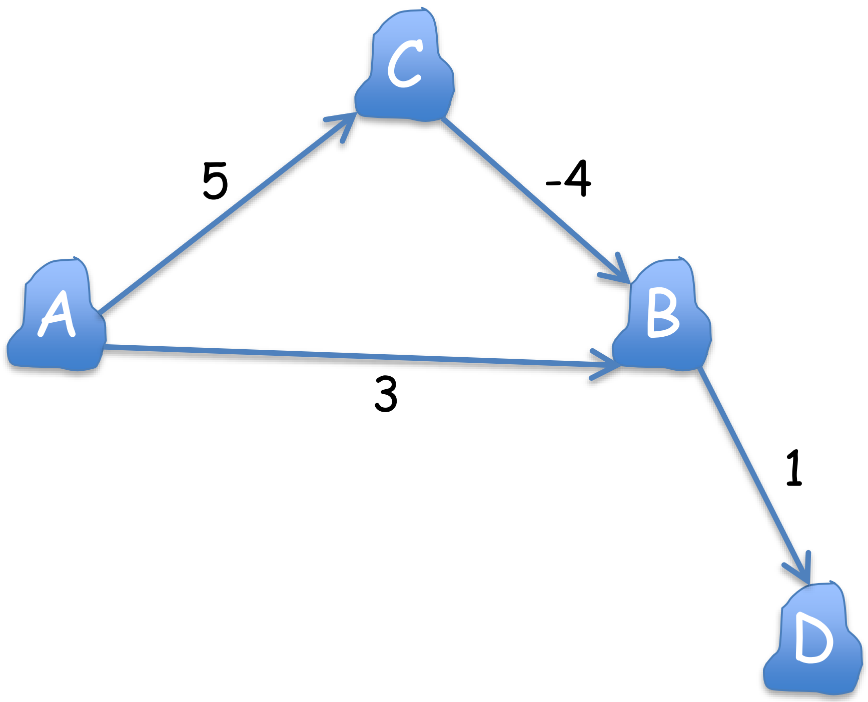
log K = $max_{P \in \{ s \rightarrow u \}}$ log $\left(\sum_{e \in P} w(e) \right)$

log (0,00011) = - 3,95

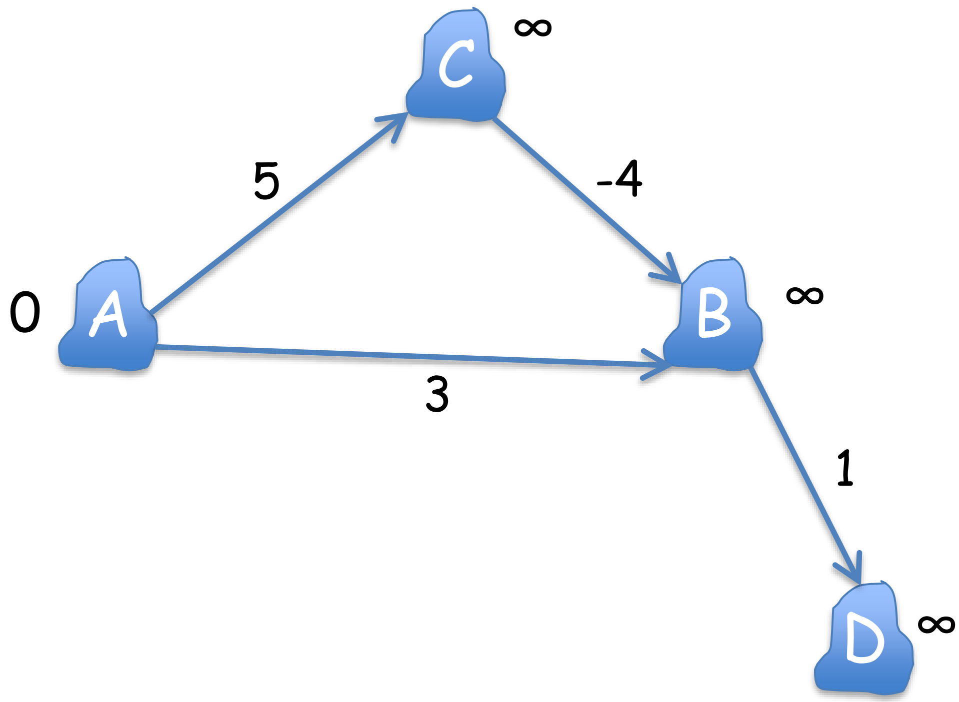
log (0,0629) = - 1,201

log (0,658) = - 0,181

Dijkstra's Algorithm

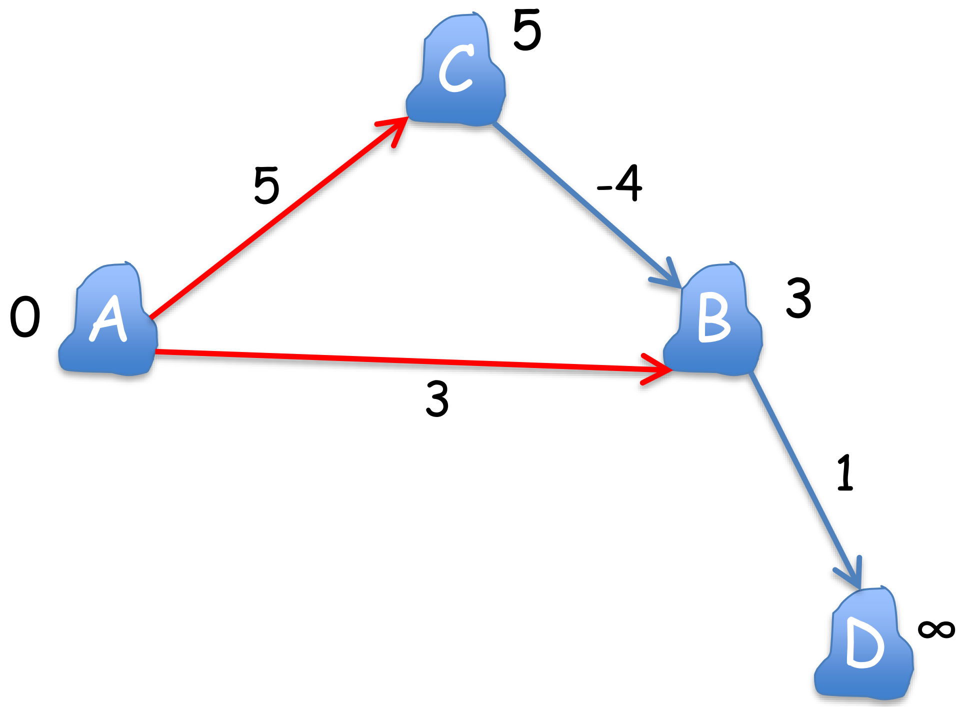


Dijkstra's Algorithm



A B C D
S = { }

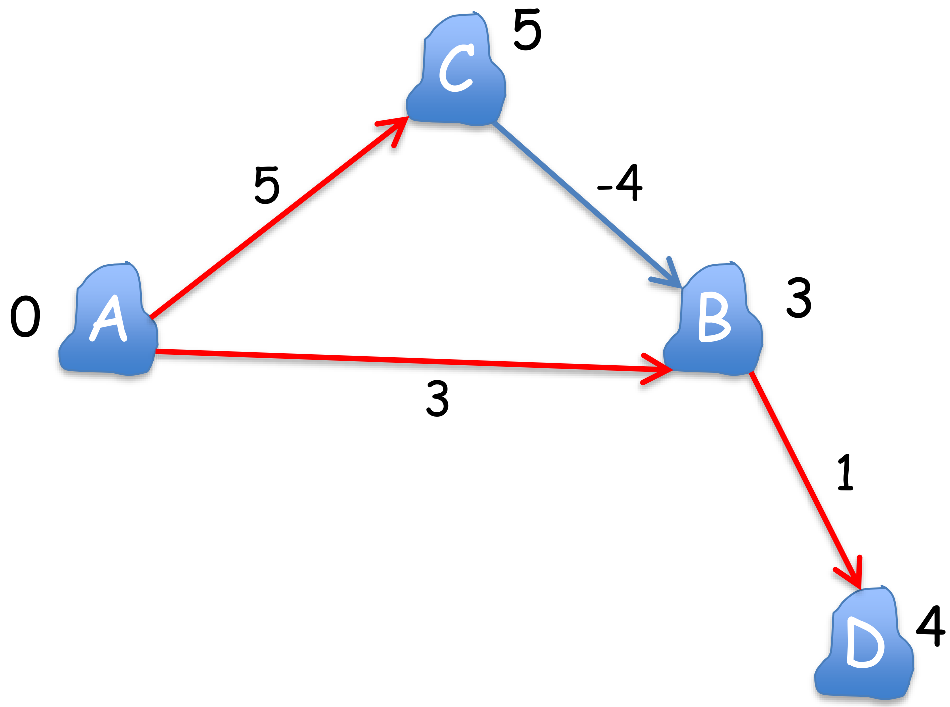
Dijkstra's Algorithm



B C D

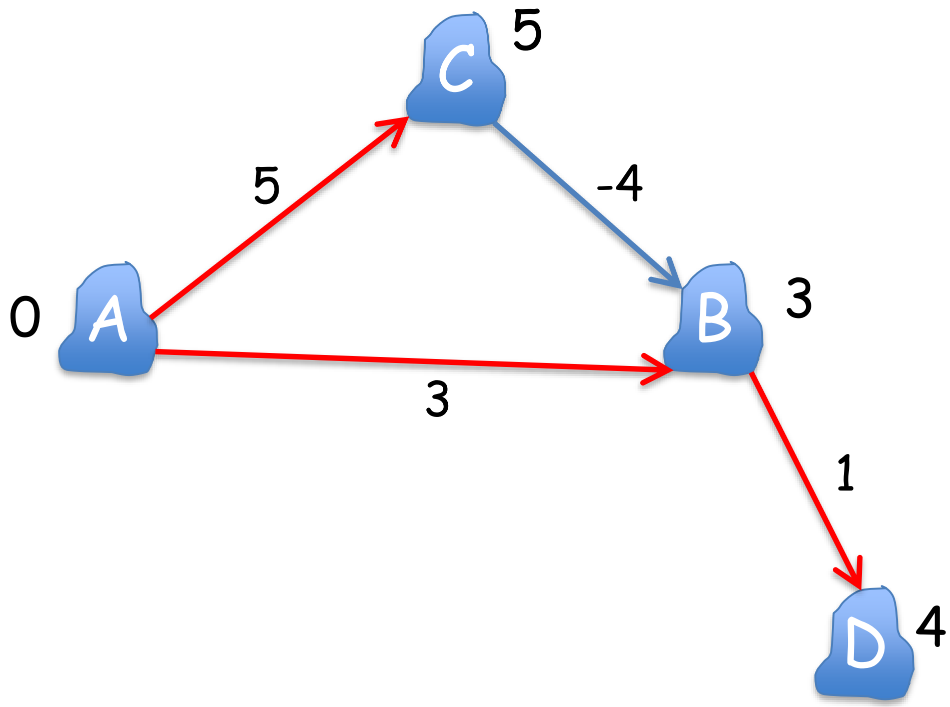
S = {A}

Dijkstra's Algorithm



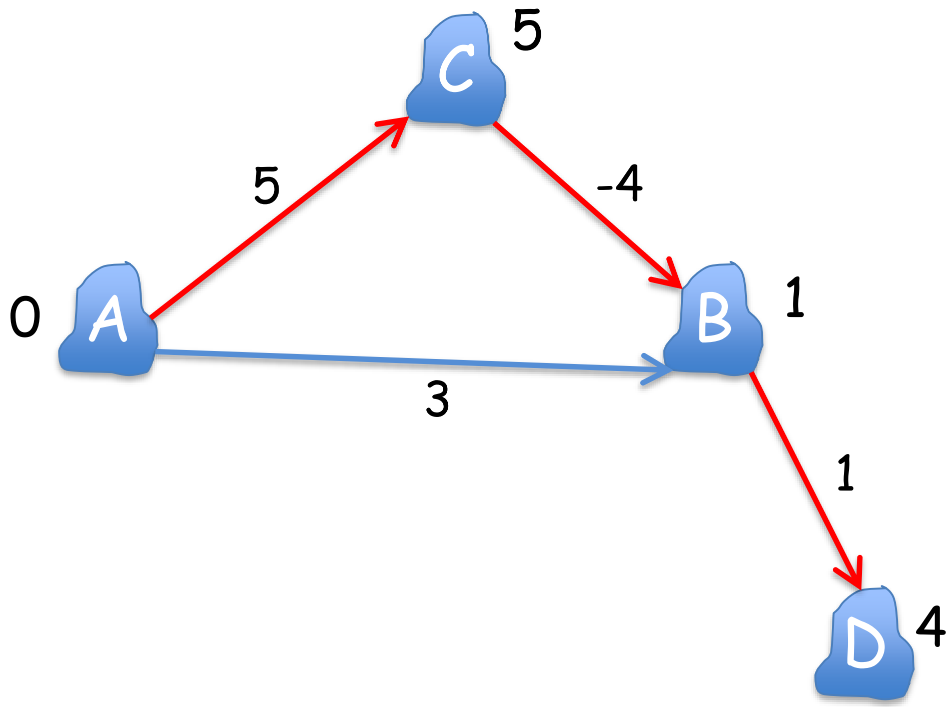
C D
S = {A B}

Dijkstra's Algorithm



C
S = {A B D}

Dijkstra's Algorithm



S = {A B D C}

The algorithm outputs 4 as the weight of the shortest path from A to D. However, the weight of the shortest path is 2.

Dijkstra's Algorithm fails for this type of problem!

Dijkstra's Algorithm

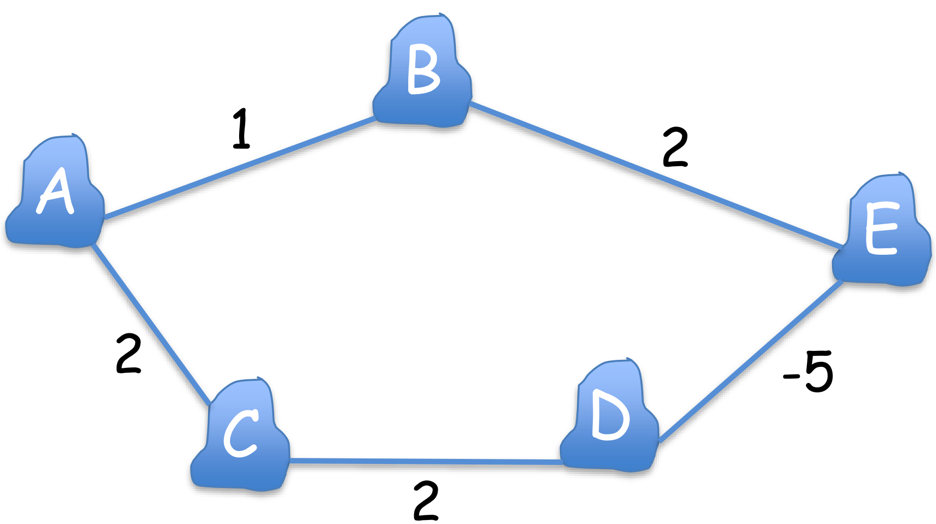
A Naive Approach

- find the lightest edge of the graph
- add the weight of that edge to all edges of the graph in order to make them non-negative
- apply Dijkstra to find the shortest paths

Dijkstra's Algorithm

A Naive Approach

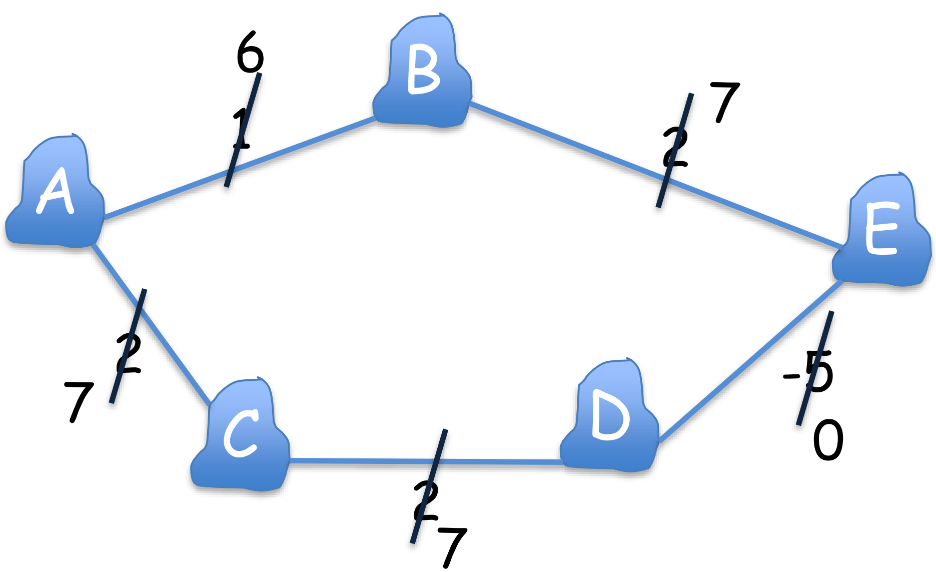
- find the lightest edge of the graph
- add the weight of that edge to all edges of the graph in order to make them non-negative
- apply Dijkstra to find the shortest paths



Dijkstra's Algorithm

A Naive Approach

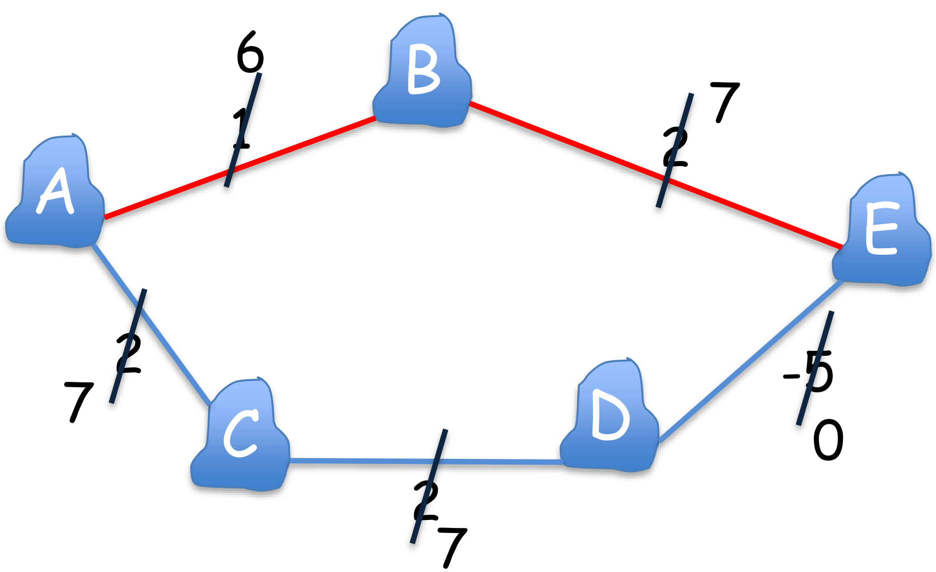
- find the lightest edge of the graph
- add the weight of that edge to all edges of the graph in order to make them non-negative
- apply Dijkstra to find the shortest paths



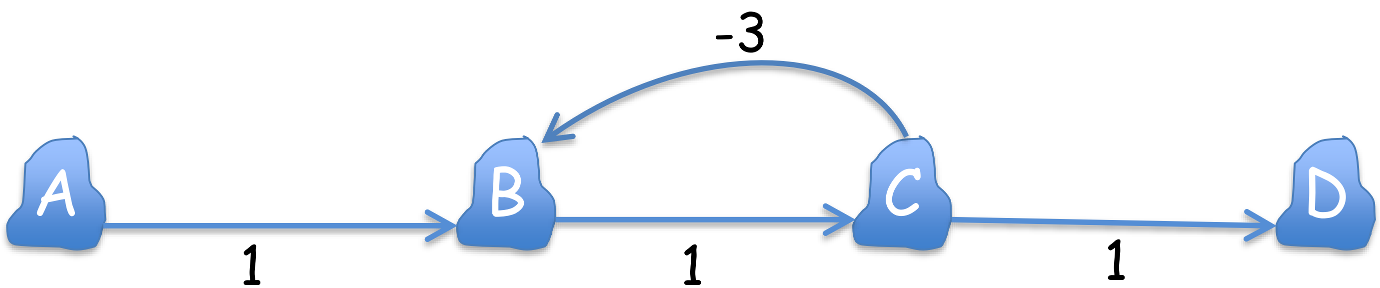
Dijkstra's Algorithm

A Naive Approach

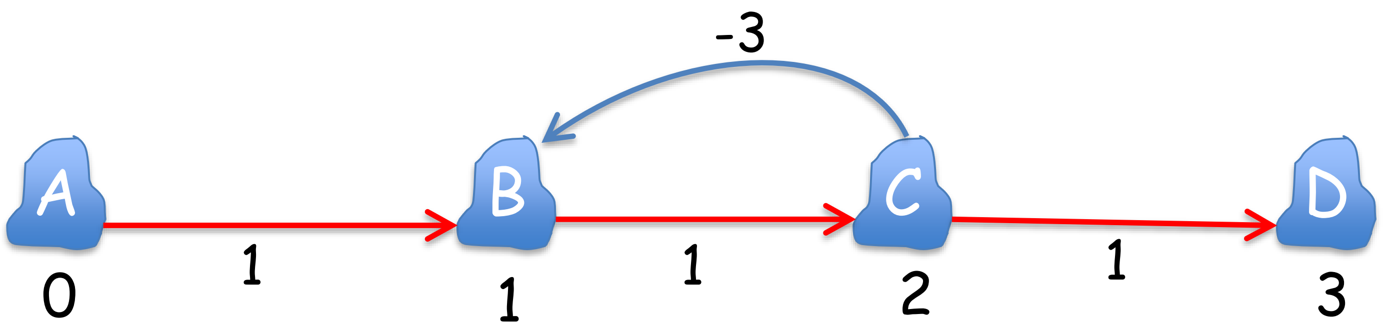
- find the lightest edge of the graph
- add the weight of that edge to all edges of the graph in order to make them non-negative
- apply Dijkstra to find the shortest paths



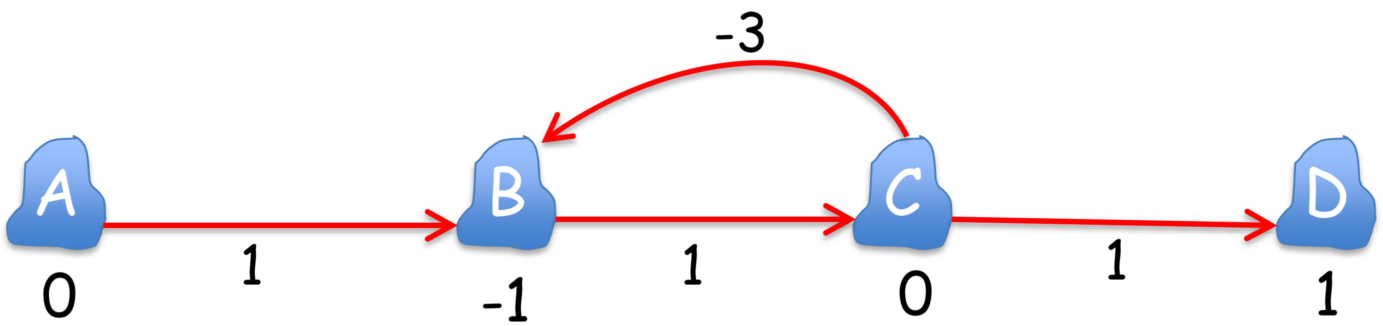
Dijkstra's Algorithm



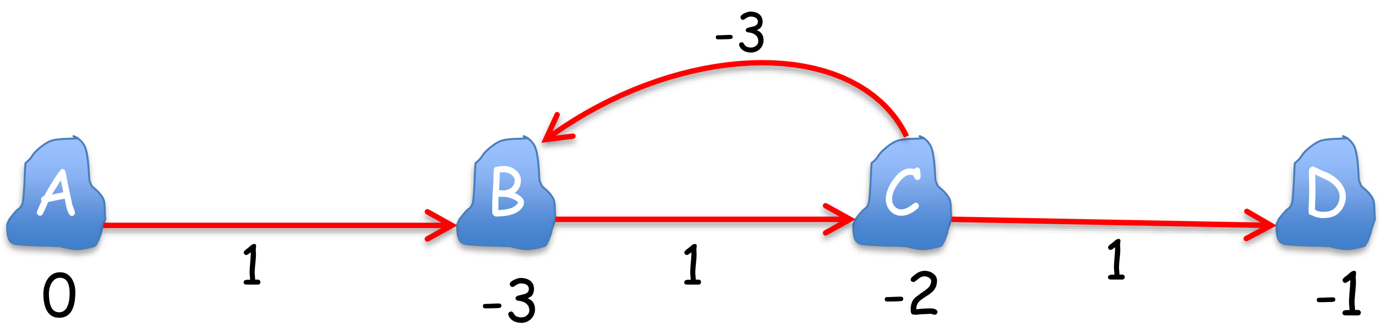
Dijkstra's Algorithm



Dijkstra's Algorithm



Dijkstra's Algorithm



Bellman-Ford

- define a subproblem

Bellman-Ford

- define a subproblem

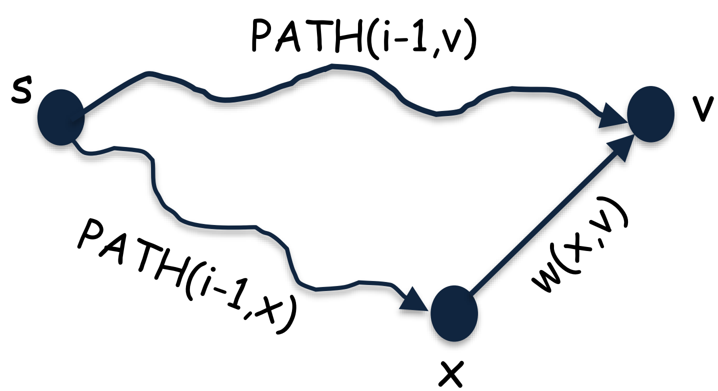
PATH(i,v) : the weight of the shortest path to v that contains $\leq i$ edges

Bellman-Ford

- define a subproblem

PATH(i,v) : the weight of the shortest path to v that contains $\leq i$ edges

- construct recurrence relation



Two Cases

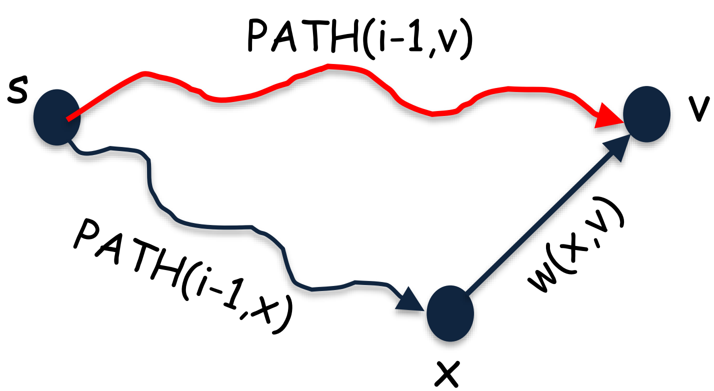
1)

Bellman-Ford

- define a subproblem

PATH(i,v) : the weight of the shortest path to v that contains $\leq i$ edges

- construct recurrence relation



Two Cases

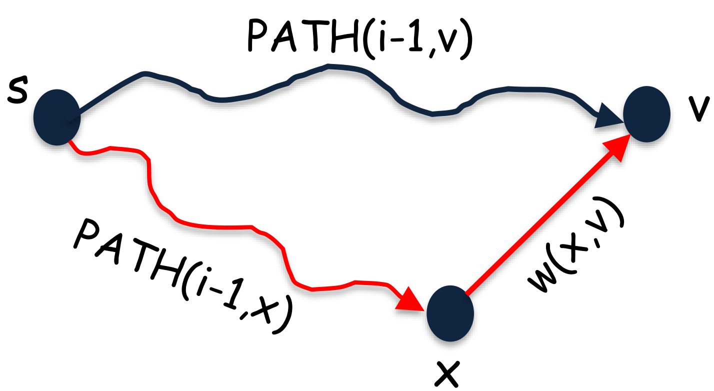
1) PATH(i-1,v)

Bellman-Ford

- define a subproblem

PATH(i,v) : the weight of the shortest path to v that contains $\leq i$ edges

- construct recurrence relation



Two Cases

relaxation

1) PATH(i-1,v)

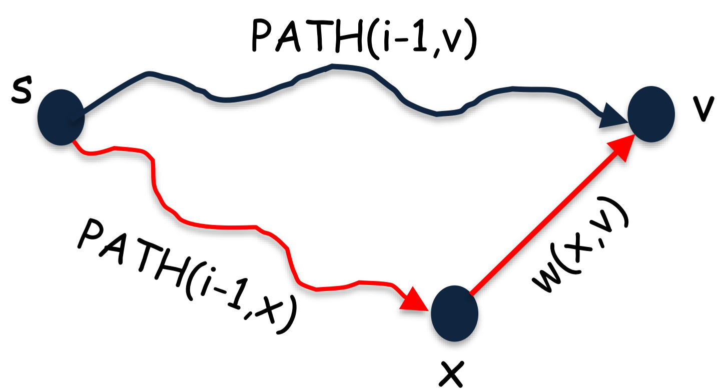
2) PATH(i-1,x) + w(x,v)

Bellman-Ford

- define a subproblem

  PATH(i,v) : the weight of the shortest path to v that contains $\leq i$ edges

- construct recurrence relation



Two Cases

relaxation

1) PATH(i-1,v)

2) PATH(i-1,x) + w(x,v)

$$PATH(i,v) = \begin{cases} \infty & , \quad i = 0 \\ 0 & , \quad s = v \\ \min\{PATH(i-1,v),\ PATH(i-1,x) + w(x,v)\} & , \text{ otherwise} \end{cases}$$

Bellman-Ford

Bellman-Ford(G,s)

```
for each u of V
      PATH(0,v) = ∞
PATH(0,s) = 0
for i = 1 to |V| -1
    for each edge (u,v) in E
          PATH(i,v) = min {PATH(i-1,v),
                           PATH(i-1,u)+w(u,v)}
```

Bellman-Ford

Bellman-Ford(G,s)

for each u of V
    PATH(0,v) = ∞
PATH(0,s) = 0
for i = 1 to |V| -1
    for each edge (u,v) in E
        PATH(i,v) = min {PATH(i-1,v),
                         PATH(i-1,u)+w(u,v)}

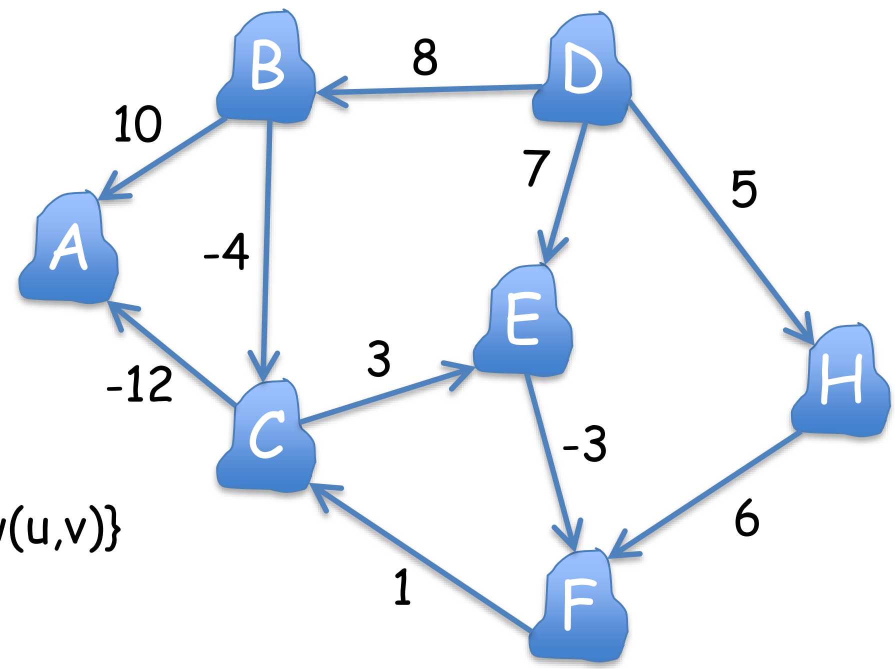
Initialize(G,s)
O(|V|)

O(|E|.|V|)

Bellman-Ford

Bellman-Ford(G,s)

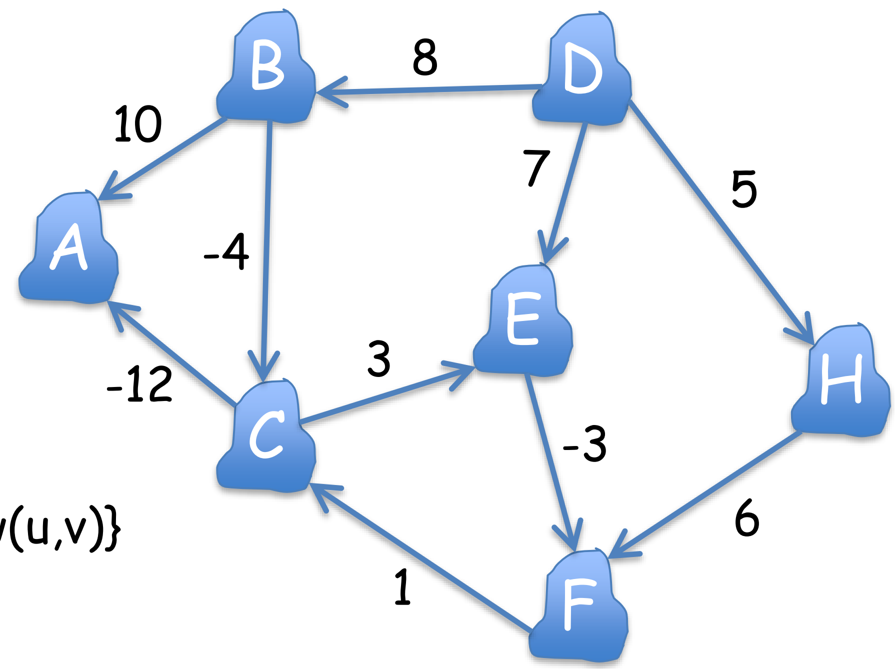
for each u of V
    PATH(0,v) = ∞
PATH(0,s) = 0
for i = 1 to |V| -1
    for each edge (u,v) in E
        PATH(i,v) = min {PATH(i-1,v),
                                    PATH(i-1,u)+w(u,v)}



Bellman-Ford

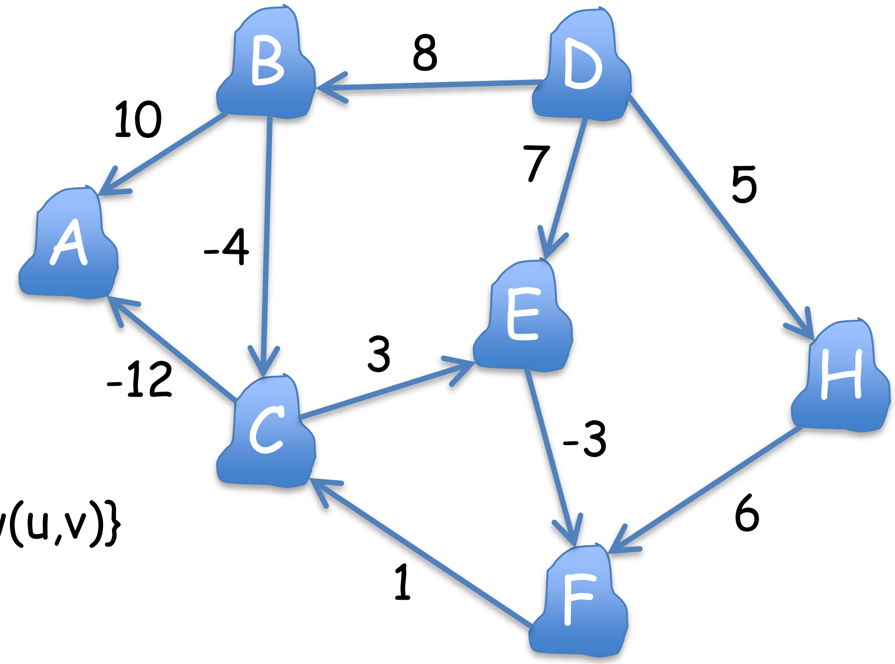
Bellman-Ford(G,s)

```
for each u of V
        PATH(0,v) = ∞
PATH(0,s) = 0
for i = 1 to |V| -1
      for each edge (u,v) in E
            PATH(i,v) = min {PATH(i-1,v),
                              PATH(i-1,u)+w(u,v)}
```



	0	1	2
A	∞		
B	∞		
C	∞		
D	0		
E	∞		
F	∞		
H	∞		

Bellman-Ford



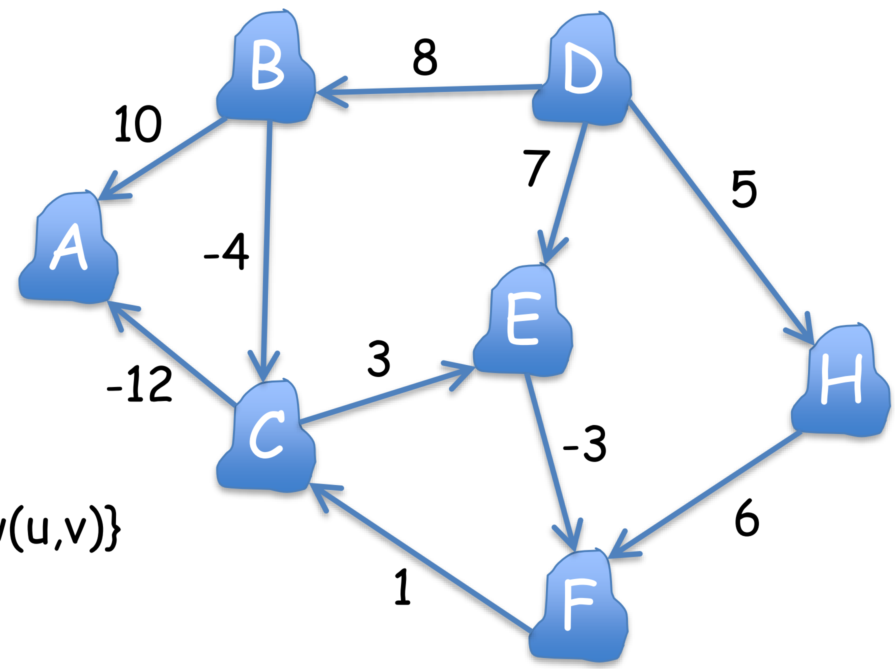
Bellman-Ford(G,s)

for each u of V
    PATH(0,v) = ∞
PATH(0,s) = 0
for i = 1 to |V| -1
    for each edge (u,v) in E
        PATH(i,v) = min {PATH(i-1,v),
                         PATH(i-1,u)+w(u,v)}

	0	1	2
A	∞	∞	
B	∞		
C	∞		
D	0		
E	∞		
F	∞		
H	∞		

$$PATH(1,A) = \min \begin{cases} PATH(0,A) \\ PATH(0,B) + w(B,A) \\ PATH(0,C) + w(C,A) \end{cases}$$

Bellman-Ford



<u>Bellman-Ford(G,s)</u>

```
for each u of V
      PATH(0,v) = ∞
PATH(0,s) = 0
for i = 1 to |V| -1
    for each edge (u,v) in E
          PATH(i,v) = min {PATH(i-1,v),
                           PATH(i-1,u)+w(u,v)}
```

	0	1	2
A	∞	∞	
B	∞	8	
C	∞		
D	0		
E	∞		
F	∞		
H	∞		

$$PATH(1,A) = \min \begin{cases} PATH(0,A) \\ PATH(0,B) + w(B,A) \\ PATH(0,C) + w(C,A) \end{cases}$$

$$PATH(1,B) = \min \begin{cases} PATH(0,B) \\ PATH(0,D) + w(D,B) \end{cases}$$

Bellman-Ford

Bellman-Ford(G,s)

for each u of V

    PATH(0,v) = ∞

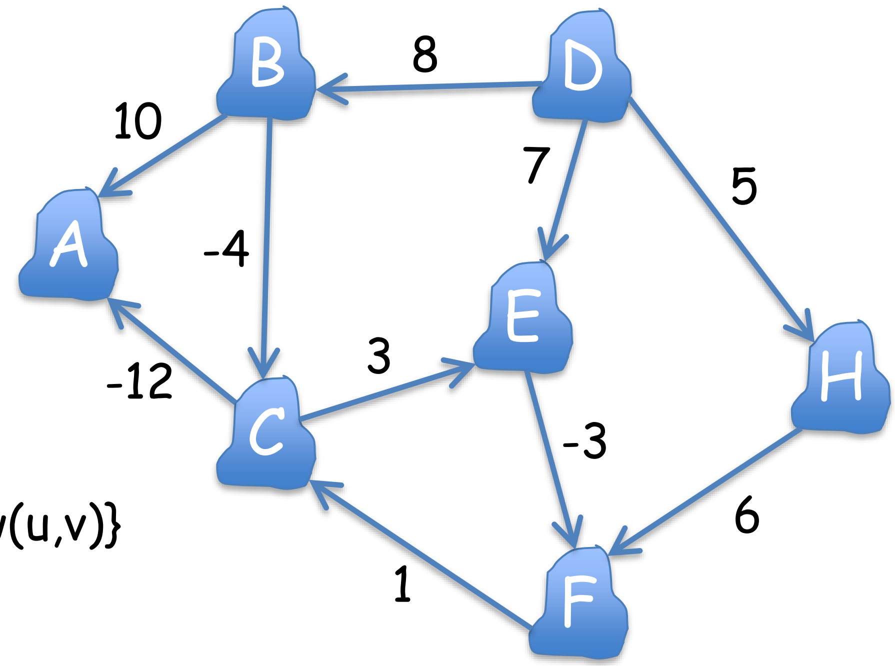
PATH(0,s) = 0

for i = 1 to |V| -1

    for each edge (u,v) in E

        PATH(i,v) = min {PATH(i-1,v),

            PATH(i-1,u)+w(u,v)}



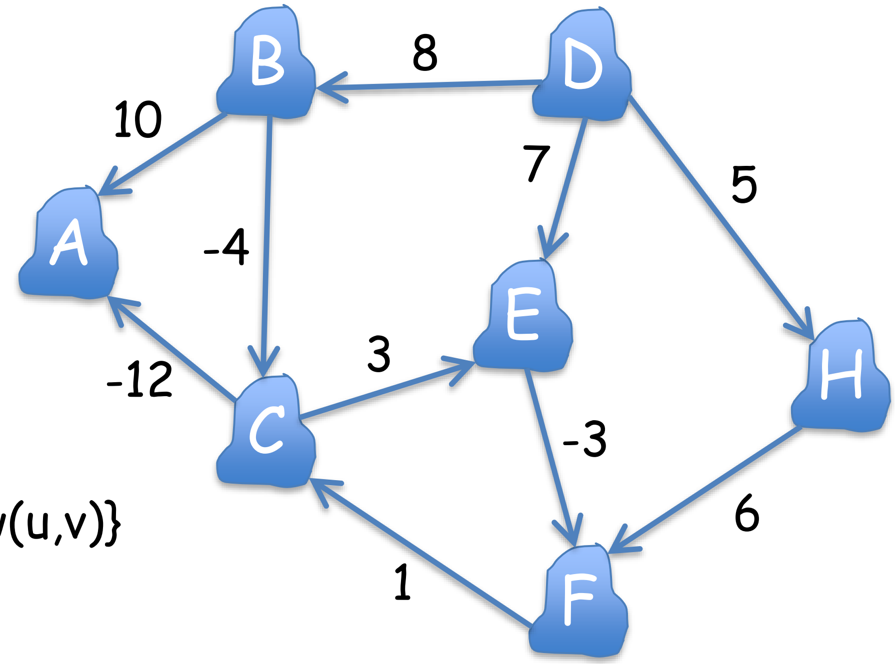
	0	1	2
A	∞	∞	18
B	∞	8	
C	∞	∞	
D	0	0	
E	∞	7	
F	∞	∞	
H	∞	5	

$$\text{PATH}(2,A) = \min \begin{cases} \text{PATH}(1,A) \\ \text{PATH}(1,B) + w(B,A) \\ \text{PATH}(1,C) + w(C,A) \end{cases}$$

Bellman-Ford

Bellman-Ford(G,s)

for each u of V
    PATH(0,v) = ∞
PATH(0,s) = 0
for i = 1 to |V| -1
    for each edge (u,v) in E
        PATH(i,v) = min {PATH(i-1,v),
                                  PATH(i-1,u)+w(u,v)}

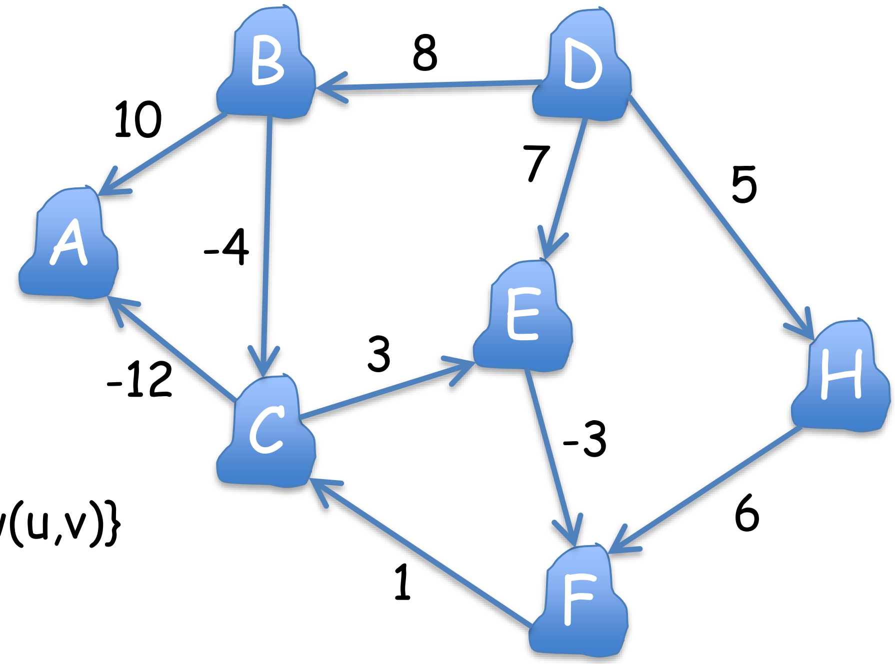


	0	1	2
A	∞	∞	18
B	∞	8	
C	∞	∞	
D	0	0	
E	∞	7	
F	∞	∞	4
H	∞	5	

$$PATH(2,A) = \min \begin{cases} PATH(1,A) \\ PATH(1,B) + w(B,A) \\ PATH(1,C) + w(C,A) \end{cases}$$

$$PATH(2,F) = \min \begin{cases} PATH(1,F) \\ PATH(1,E) + w(E,F) \\ PATH(1,H) + w(H,F) \end{cases}$$

Bellman-Ford



Bellman-Ford(G,s)

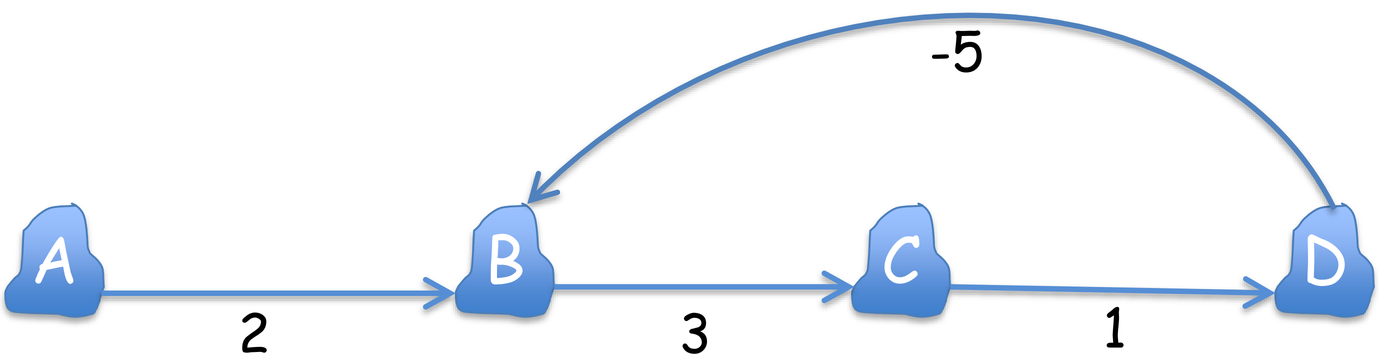
for each u of V
  PATH(0,v) = ∞
PATH(0,s) = 0
for i = 1 to |V| -1
  for each edge (u,v) in E
    PATH(i,v) = min {PATH(i-1,v),
      PATH(i-1,u)+w(u,v)}

	0	1	2
A	∞	∞	18
B	∞	8	8
C	∞	∞	4
D	0	0	0
E	∞	7	7
F	∞	∞	4
H	∞	5	5

Bellman-Ford

<u>Bellman-Ford(G,s)</u>

```
for each u of V
      PATH(0,v) = ∞
PATH(0,s) = 0
for i = 1 to |V| -1
     for each edge (u,v) in E
          PATH(i,v) = min {PATH(i-1,v),
                           PATH(i-1,u)+w(u,v)}
```



Bellman-Ford

Bellman-Ford(G,s)

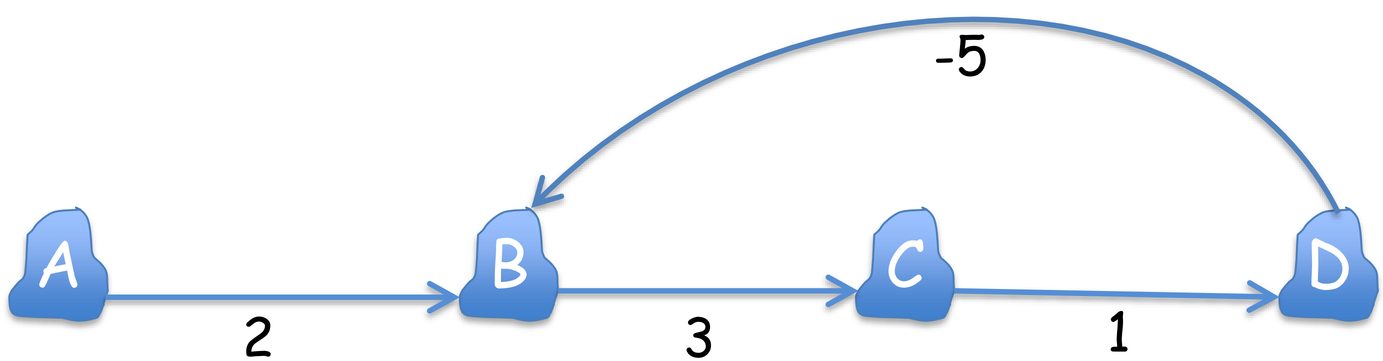
```
for each u of V
      PATH(0,v) = ∞
PATH(0,s) = 0
for i = 1 to |V| -1
     for each edge (u,v) in E
          PATH(i,v) = min {PATH(i-1,v),
                          PATH(i-1,u)+w(u,v)}
```

- the shortest path contains at most |V| -1 edges.
- more than this creates a cycle
  (use this fact to detect a negative cycle)

Bellman-Ford

Bellman-Ford(G,s)

for each u of V
    PATH(0,v) = ∞
PATH(0,s) = 0
for i = 1 to |V| -1
    for each edge (u,v) in E
        PATH(i,v) = min {PATH(i-1,v),
                         PATH(i-1,u)+w(u,v)}

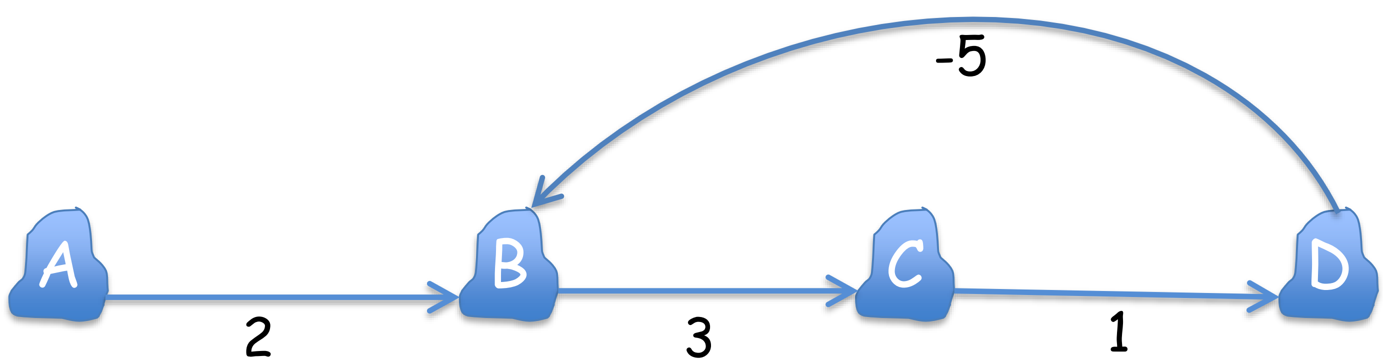


Bellman-Ford

Bellman-Ford(G,s)

for each u of V
    PATH(0,v) = ∞
PATH(0,s) = 0
for i = 1 to |V| -1
    for each edge (u,v) in E
        PATH(i,v) = min {PATH(i-1,v),
                         PATH(i-1,u)+w(u,v)}

	0	1	2	3
A	0	0	0	0
B	∞	2	2	2
C	∞	∞	5	5
D	∞	∞	∞	6

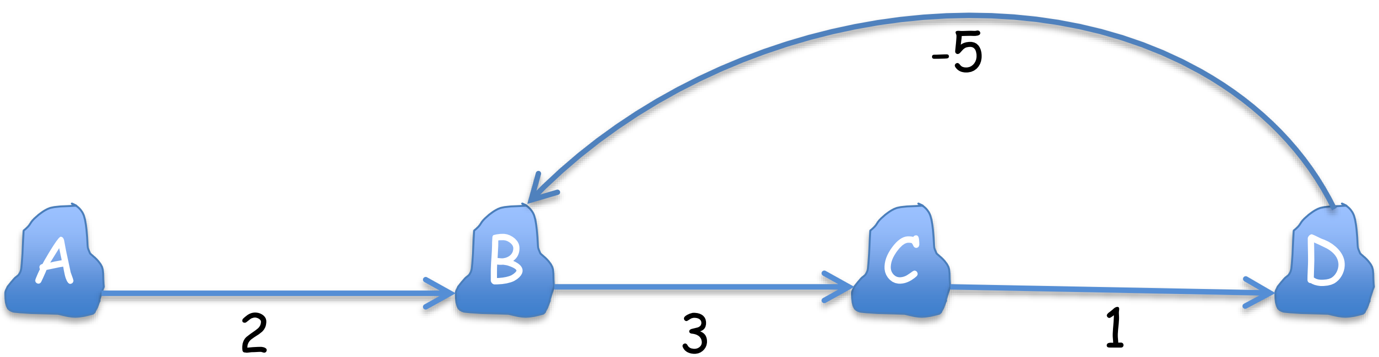


Bellman-Ford

Bellman-Ford(G,s)

```
for each u of V
        PATH(0,v) = ∞
PATH(0,s) = 0
for i = 1 to |V| -1
      for each edge (u,v) in E
              PATH(i,v) = min {PATH(i-1,v),
                                PATH(i-1,u)+w(u,v)}
```

	0	1	2	3		
A	0	0	0	0	0	0
B	∞	2	2	2	1	1
C	∞	∞	5	5	5	4
D	∞	∞	∞	6	6	6



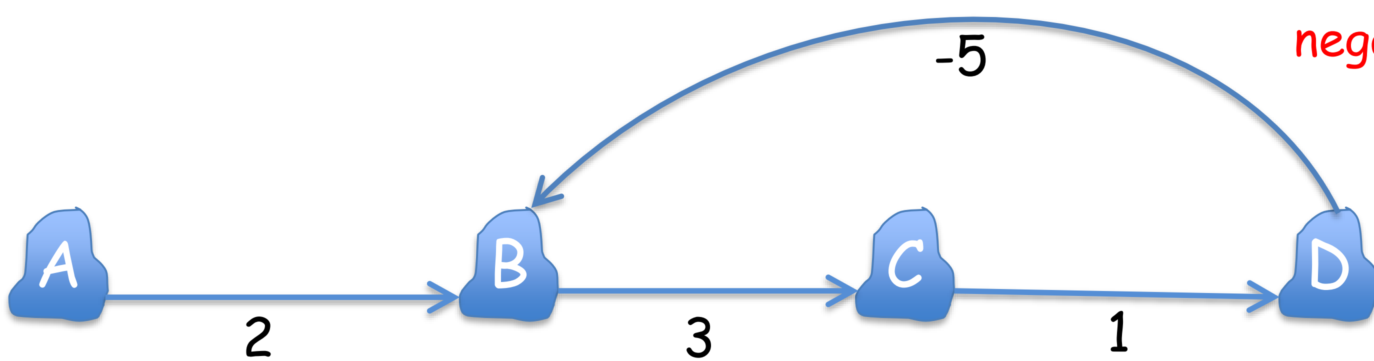
Bellman-Ford

Bellman-Ford(G,s)

```
for each u of V
        PATH(0,v) = ∞
PATH(0,s) = 0
for i = 1 to |V| -1
      for each edge (u,v) in E
            PATH(i,v) = min {PATH(i-1,v),
                             PATH(i-1,u)+w(u,v)}
```

	0	1	2	3		
A	0	0	0	0	0	0
B	∞	2	2	2	1	1
C	∞	∞	5	5	5	4
D	∞	∞	∞	6	6	6

negative cycle!



APSP

- given a weighted graph $G=(V,E)$ and a source vertex s in V, find the shortest paths from each vertex to every other vertex in V

APSP

- given a weighted graph G=(V,E) and a source vertex s in V, find the shortest paths from each vertex to every other vertex in V

- If it does not contain negative weight edge, run Dijkstra's Algorithm on each vertex

- If it contains negative weight edge, run Bellman-Ford on each vertex

APSP

- given a weighted graph G=(V,E) and a source vertex s in V, find the shortest paths from each vertex to every other vertex in V

- If it does not contain negative weight edge, run Dijkstra's Algorithm on each vertex

  O(|V|.|E|.log|V|)

  (for dense graph $\approx O(|V|^3 \log|V|)$)

- If it contains negative weight edge, run Bellman-Ford on each vertex

  $O(|V|^2|E|)$

  (for dense graph $\approx O(|V|^4)$)

APSP

- define a subproblem

Suppose the graph given with adjacency matrix of weight :

$W = (w_{ij})$ such that $w_{ij} = \begin{cases} 0, \ if \ i = j \\ \infty, \ \ if \ i \neq j \ and \ (i,j) \ not \ in \ E \\ w(i,j), \ if i \neq j \ and \ (i,j) \ in \ E \end{cases}$

APSP

- define a subproblem

$PATH_{ij}^{(m)}$: the weight of the shortest path from i to j that contains $\leq m$ edges

APSP

- define a subproblem

$PATH_{ij}^{(m)}$: the weight of the shortest path from i to j that contains $\leq m$ edges

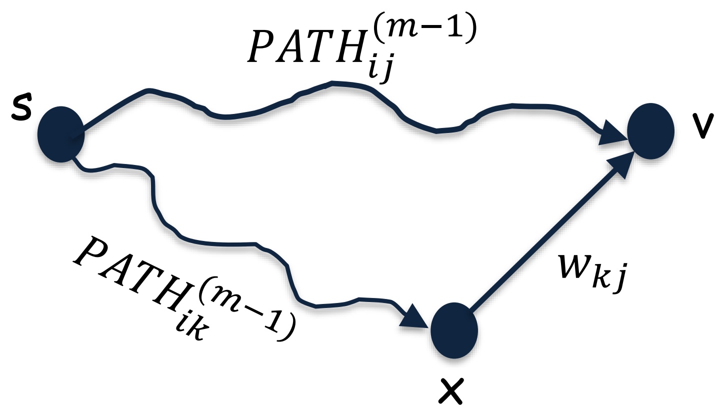
- construct recurrence relation

APSP

- define a subproblem

$PATH_{ij}^{(m)}$: the weight of the shortest path from i to j that contains $\leq m$ edges

- construct recurrence relation



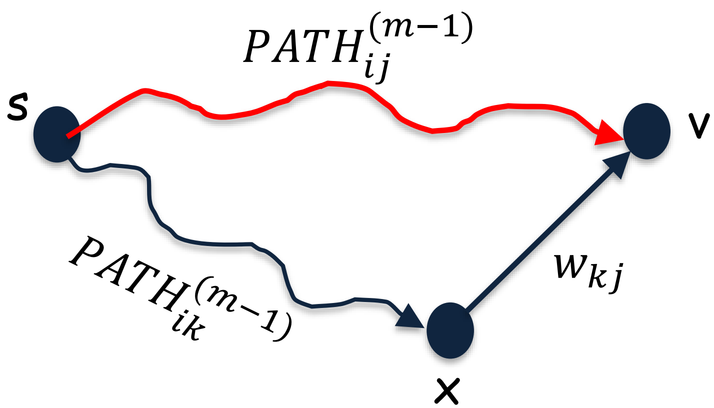
Two Cases

APSP

- define a subproblem

$PATH_{ij}^{(m)}$: the weight of the shortest path from i to j that contains $\leq m$ edges

- construct recurrence relation



Two Cases

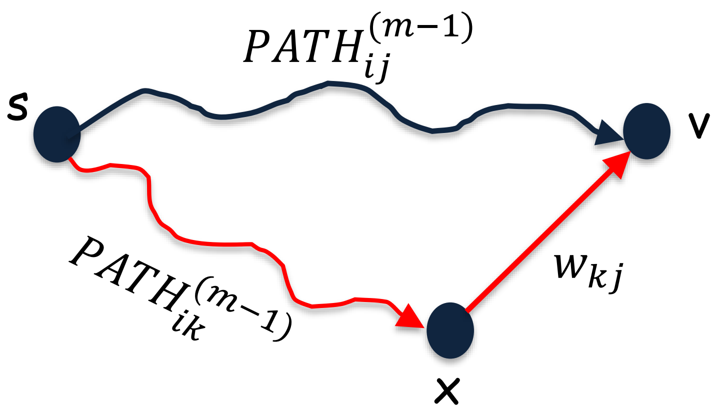
1) $PATH_{ij}^{(m-1)}$

APSP

- define a subproblem

$PATH_{ij}^{(m)}$: the weight of the shortest path from i to j that contains $\leq m$ edges

- construct recurrence relation



Two Cases

1) $PATH_{ij}^{(m-1)}$

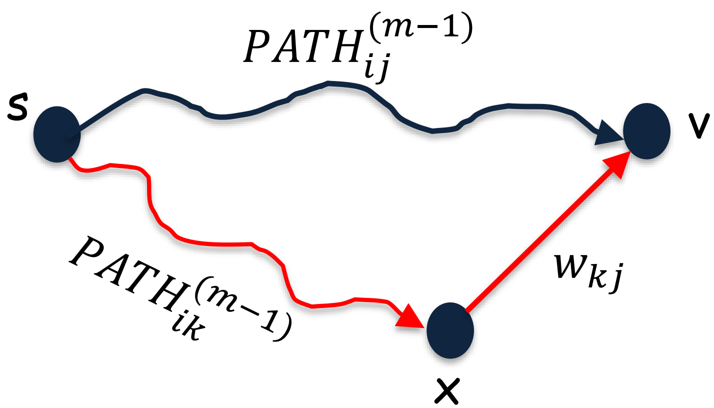
2) $PATH_{ik}^{(m-1)} + w_{kj}$

APSP

- define a subproblem

$PATH_{ij}^{(m)}$: the weight of the shortest path from i to j that contains $\leq m$ edges

- construct recurrence relation



Two Cases

1) $PATH_{ij}^{(m-1)}$

2) $PATH_{ik}^{(m-1)} + w_{kj}$

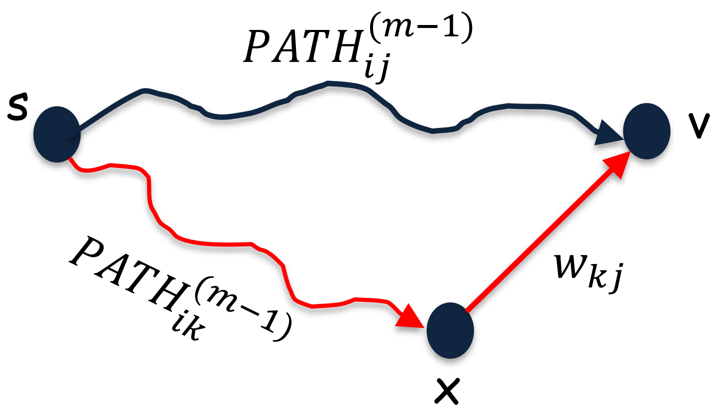
$$PATH_{ij}^{(m)} = \min\left\{PATH_{ij}^{(m-1)}, min_{1\leq k\leq n}\left\{PATH_{ik}^{(m-1)} + w_{kj}\right\}\right\}$$

APSP

- define a subproblem

$PATH_{ij}^{(m)}$: the weight of the shortest path from i to j that contains $\le m$ edges

- construct recurrence relation



Two Cases

1) $PATH_{ij}^{(m-1)}$

2) $PATH_{ik}^{(m-1)} + w_{kj}$

$$PATH_{ij}^{(m)} = \min\left\{PATH_{ij}^{(m-1)}, min_{1\le k\le n}\left\{PATH_{ik}^{(m-1)} + w_{kj}\right\}\right\}$$

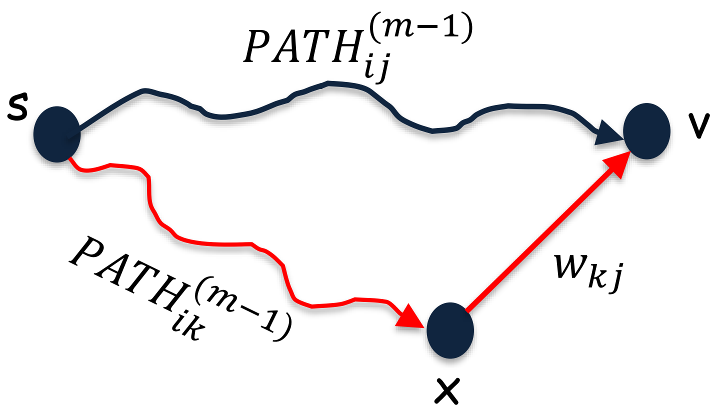
it's contained in this part

APSP

- define a subproblem

$PATH_{ij}^{(m)}$: the weight of the shortest path from i to j that contains $\leq m$ edges

- construct recurrence relation



Two Cases

1) $PATH_{ij}^{(m-1)}$

2) $PATH_{ik}^{(m-1)} + w_{kj}$

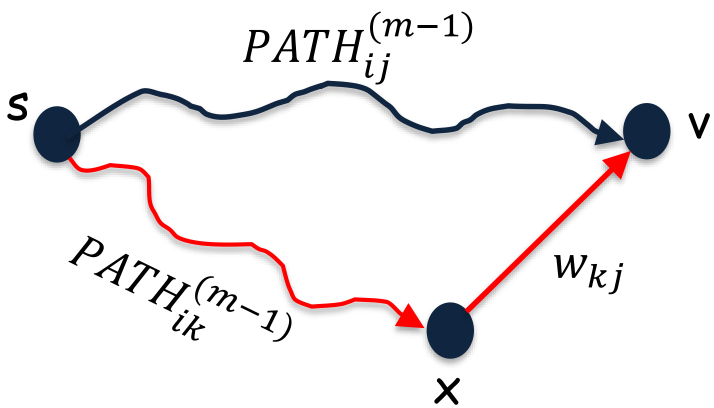
$$PATH_{ij}^{(m)} = min_{1 \leq k \leq n} \left\{ PATH_{ik}^{(m-1)} + w_{kj} \right\}$$

APSP

- define a subproblem

$PATH_{ij}^{(m)}$: the weight of the shortest path from i to j that contains $\leq m$ edges

- construct recurrence relation



Two Cases

1) $PATH_{ij}^{(m-1)}$

2) $PATH_{ik}^{(m-1)} + w_{kj}$

$$PATH_{ij}^{(m)} = min_{1 \leq k \leq n} \left\{ PATH_{ik}^{(m-1)} + w_{kj} \right\}$$

$$PATH_{ii}^{(0)} = 0 \qquad PATH_{ij}^{(0)} = \infty$$

APSP

- Let's define a matrix

$$P^{(m)} = PATH_{ij}^{(m)}$$

APSP

- Let's define a matrix

$$P^{(m)} = PATH_{ij}^{(m)}$$

- $P^{(1)} = W$ since $w_{ij} = PATH_{ij}^{(1)}$

APSP

- Let's define a matrix

$$P^{(m)} = PATH_{ij}^{(m)}$$

- $P^{(1)} = W$ since $w_{ij} = PATH_{ij}^{(1)}$

- Then, we are computing the matrix $P^{(n-1)}$ since $PATH_{ij}^{(n-1)} = \delta(i,j)$

APSP

- Let's define a matrix

$$P^{(m)} = PATH_{ij}^{(m)}$$

- $P^{(1)} = W$ since $w_{ij} = PATH_{ij}^{(1)}$

- Then, we are computing the matrix $P^{(n-1)}$ since $PATH_{ij}^{(n-1)} = \delta(i,j)$

$$\begin{bmatrix} PATH_{11}^{(m-1)} & \cdots & PATH_{1n}^{(m-1)} \\ \vdots & \ddots & \vdots \\ PATH_{n1}^{(m-1)} & \cdots & PATH_{nn}^{(m-1)} \end{bmatrix} . \begin{bmatrix} w_{11} & \cdots & w_{1n} \\ \vdots & \ddots & \vdots \\ w_{n1} & \cdots & w_{nn} \end{bmatrix} = \begin{bmatrix} PATH_{11}^{(m)} & \cdots & PATH_{1n}^{(m)} \\ \vdots & \ddots & \vdots \\ PATH_{n1}^{(m)} & \cdots & PATH_{nn}^{(m)} \end{bmatrix}$$

$$PATH_{ij}^{(m)} = PATH_{i1}^{(m-1)}.\, w_{1j} + PATH_{i2}^{(m-1)}.\, w_{2j} + \cdots + PATH_{in}^{(m-1)}.\, w_{nj}$$

APSP

- Let's define a matrix

$$P^{(m)} = PATH_{ij}^{(m)}$$

- $P^{(1)} = W$ since $w_{ij} = PATH_{ij}^{(1)}$

- Then, we are computing the matrix $P^{(n-1)}$ since $PATH_{ij}^{(n-1)} = \delta(i,j)$

$$\begin{bmatrix} PATH_{11}^{(m-1)} & \cdots & PATH_{1n}^{(m-1)} \\ \vdots & \ddots & \vdots \\ PATH_{n1}^{(m-1)} & \cdots & PATH_{nn}^{(m-1)} \end{bmatrix} . \begin{bmatrix} w_{11} & \cdots & w_{1n} \\ \vdots & \ddots & \vdots \\ w_{n1} & \cdots & w_{nn} \end{bmatrix} = \begin{bmatrix} PATH_{11}^{(m)} & \cdots & PATH_{1n}^{(m)} \\ \vdots & \ddots & \vdots \\ PATH_{n1}^{(m)} & \cdots & PATH_{nn}^{(m)} \end{bmatrix}$$

$$PATH_{ij}^{(m)} = PATH_{i1}^{(m-1)}.\, w_{1j} + PATH_{i2}^{(m-1)}.\, w_{2j} + \cdots + PATH_{in}^{(m-1)}.\, w_{nj}$$

$$PATH_{ij}^{(m)} = min_{1 \le k \le n} \left\{ PATH_{ik}^{(m-1)} + w_{kj} \right\}$$

APSP

- The idea is to extend $P^{(1)}$ using the weight matrix W

APSP

- The idea is to extend $P^{(1)}$ using the weight matrix W

$P^{(1)}$ = W
for m = 2 to n-1

$$P^{(m)} = Extend(P^{(m-1)}, W, n)$$

return $P^{(n-1)}$

APSP

- The idea is to extend $P^{(1)}$ using the weight matrix W

$P^{(1)}$ = W
for m = 2 to n-1

$$P^{(m)} = Extend(P^{(m-1)}, W, n)$$

return $P^{(n-1)}$

Extend(P,W,n)

initialize P' as $n \times n$ matrix
for i = 1 to n
    for j = 1 to n
        $PATH\ _{ij}{}'= \infty$
        for k = 1 to n
            $PATH\ '_{ij} = \min\{PATH\ '_{ij}, PATH\ _{ij} + w\ _{ij}\}$

APSP

- The idea is to extend $P^{(1)}$ using the weight matrix W

$P^{(1)}$ = W
for m = 2 to n-1

$$P^{(m)} = Extend(P^{(m-1)}, W, n)$$

return $P^{(n-1)}$

Extend(P,W,n)

$O(n^3)$

initialize P' as $n \times n$ matrix
for i = 1 to n
for j = 1 to n
$PATH_{ij}'$ = ∞
for k = 1 to n
$$PATH'_{ij} = \min\{PATH'_{ij}, PATH_{ij} + w_{ij}\}$$

APSP

- The idea is to extend $P^{(1)}$ using the weight matrix W

$P^{(1)}$ = W
for m = 2 to n-1

$$P^{(m)} = Extend(P^{(m-1)}, W, n)$$

$O(n^4)$

return $P^{(n-1)}$

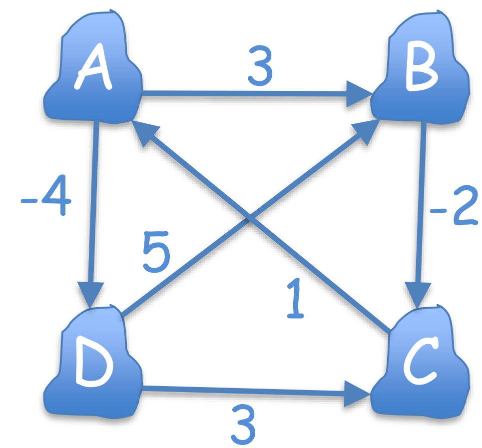
Extend(P,W,n)

$O(n^3)$

initialize P' as $n \times n$ matrix
for i = 1 to n
  for j = 1 to n
    $PATH\ _{ij}{}'$= ∞
    for k = 1 to n
      $PATH\ '_{ij} = \min\{PATH\ '_{ij}, PATH\ _{ij} + w\ _{ij}\}$

APSP

- The idea is to extend $P^{(1)}$ using the weight matrix W



$P^{(1)}$ = W
for m = 2 to n-1

$$P^{(m)} = Extend(P^{(m-1)}, W, n)$$

return $P^{(n-1)}$

<u>Extend(P,W,n)</u>

initialize P' as $n \times n$ matrix
for i = 1 to n
    for j = 1 to n
        $PATH_{ij}{}'= \infty$
        for k = 1 to n
            $PATH'_{ij} = \min\{PATH'_{ij}, PATH_{ij} + w_{ij}\}$

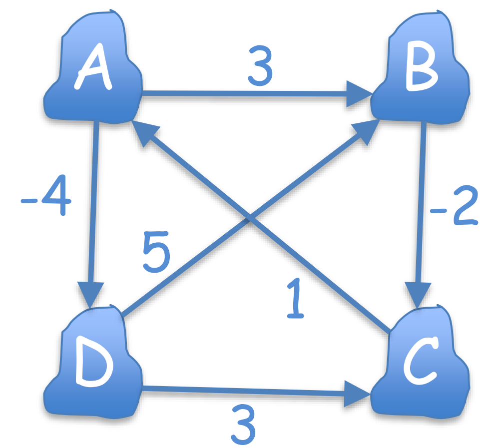
APSP

- The idea is to extend $P^{(1)}$ using the weight matrix W

$P^{(1)}$ = W
for m = 2 to n-1

$$P^{(m)} = Extend(P^{(m-1)}, W, n)$$

return $P^{(n-1)}$



$$P^{(2)} = Extend\left(P^{(1)}, W, n\right)$$

APSP

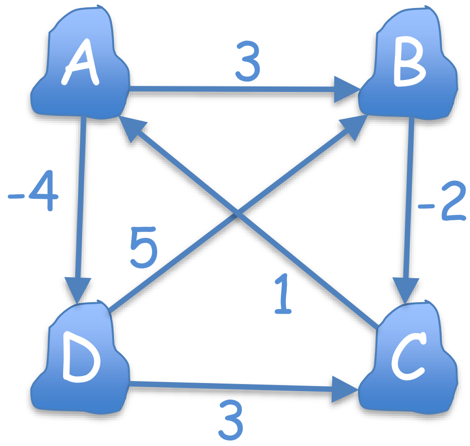
- The idea is to extend $P^{(1)}$ using the weight matrix W

$P^{(1)}$ = W
for m = 2 to n-1

$$P^{(m)} = Extend(P^{(m-1)}, W, n)$$

return $P^{(n-1)}$

$$P^{(2)} = Extend\left(P^{(1)}, W, n\right)$$
$$= P^{(1)} * W$$



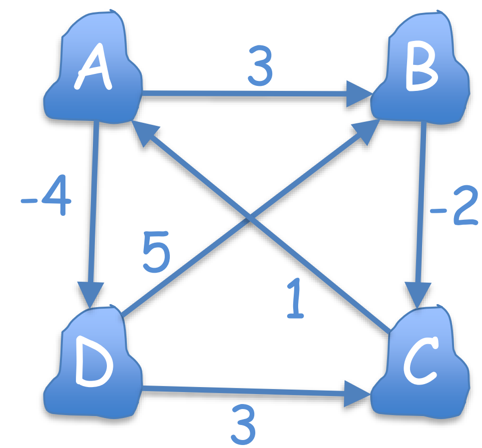
APSP

- The idea is to extend $P^{(1)}$ using the weight matrix W

$P^{(1)}$ = W
for m = 2 to n-1

$$P^{(m)} = Extend(P^{(m-1)}, W, n)$$

return $P^{(n-1)}$



$$P^{(2)} = Extend\left(P^{(1)}, W, n\right)$$
$$= P^{(1)} * W$$

$$P^{(2)} = \begin{pmatrix} 0 & 3 & \infty & -4 \\ \infty & 0 & -2 & \infty \\ 1 & \infty & 0 & \infty \\ \infty & 5 & 3 & 0 \end{pmatrix} \begin{pmatrix} 0 & 3 & \infty & -4 \\ \infty & 0 & -2 & \infty \\ 1 & \infty & 0 & \infty \\ \infty & 5 & 3 & 0 \end{pmatrix}$$

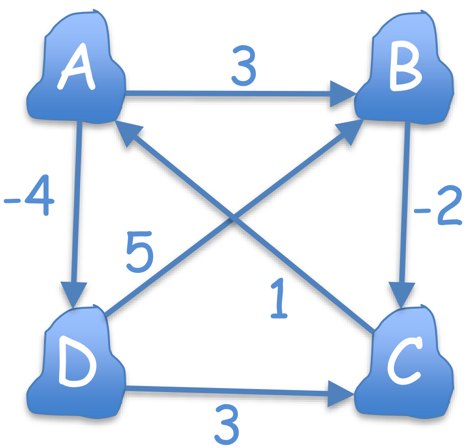
APSP

- The idea is to extend $P^{(1)}$ using the weight matrix W

$P^{(1)}$ = W
for m = 2 to n-1

$$P^{(m)} = Extend(P^{(m-1)}, W, n)$$

return $P^{(n-1)}$



$$P^{(2)} = Extend\left(P^{(1)}, W, n\right)$$
$$= P^{(1)} * W$$

$$\boxed{PATH\ '_{ij} = \min\{PATH\ '_{ij}, PATH\ _{ij} + w\ _{ij}\}}$$

$$P^{(2)} = \begin{pmatrix} 0 & 3 & \infty & -4 \\ \infty & 0 & -2 & \infty \\ 1 & \infty & 0 & \infty \\ \infty & 5 & 3 & 0 \end{pmatrix} \begin{pmatrix} 0 & 3 & \infty & -4 \\ \infty & 0 & -2 & \infty \\ 1 & \infty & 0 & \infty \\ \infty & 5 & 3 & 0 \end{pmatrix} = \begin{pmatrix} 0 & & & \\ & & & \\ & & & \\ & & & \end{pmatrix}$$

min{0+0,3+∞, ∞+1,-4+ ∞}

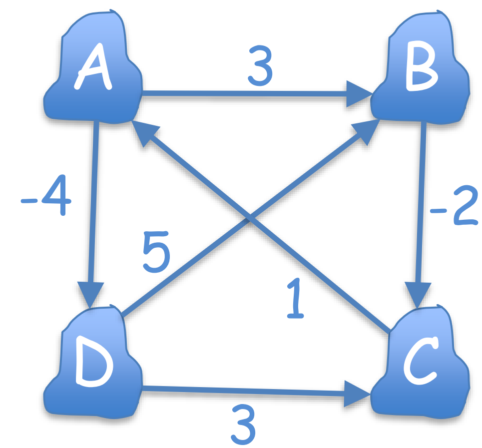
APSP

- The idea is to extend $P^{(1)}$ using the weight matrix W

$P^{(1)}$ = W
for m = 2 to n-1

$$P^{(m)} = Extend(P^{(m-1)}, W, n)$$

return $P^{(n-1)}$



$$P^{(2)} = Extend\left(P^{(1)}, W, n\right)$$
$$= P^{(1)} * W$$

$$PATH'_{ij} = \min\{PATH'_{ij}, PATH_{ij} + w_{ij}\}$$

$$P^{(2)} = \begin{pmatrix} 0 & 3 & \infty & -4 \\ \infty & 0 & -2 & \infty \\ 1 & \infty & 0 & \infty \\ \infty & 5 & 3 & 0 \end{pmatrix} \begin{pmatrix} 0 & 3 & \infty & -4 \\ \infty & 0 & -2 & \infty \\ 1 & \infty & 0 & \infty \\ \infty & 5 & 3 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 1 & & \\ & & & \\ & & & \\ & & & \end{pmatrix}$$

min{0+3,3+0, ∞+∞,-4+5}

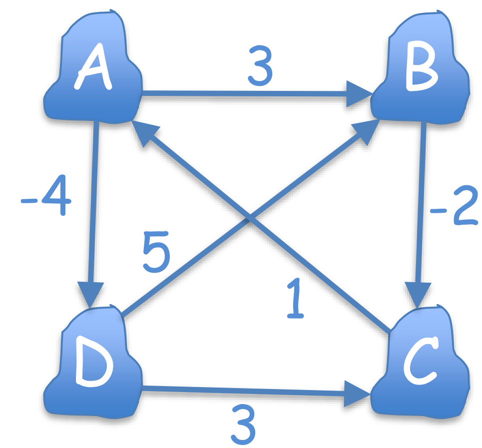
APSP

- The idea is to extend $P^{(1)}$ using the weight matrix W

$P^{(1)}$ = W
for m = 2 to n-1

$$P^{(m)} = Extend(P^{(m-1)}, W, n)$$

return $P^{(n-1)}$



$$P^{(2)} = Extend\left(P^{(1)}, W, n\right)$$
$$= P^{(1)} * W$$

$$PATH\ '_{ij} = \min\{PATH\ '_{ij}, PATH\ _{ij} + w\ _{ij}\}$$

$$P^{(2)} = \begin{pmatrix} 0 & 3 & \infty & -4 \\ \infty & 0 & -2 & \infty \\ 1 & \infty & 0 & \infty \\ \infty & 5 & 3 & 0 \end{pmatrix} \begin{pmatrix} 0 & 3 & \infty & -4 \\ \infty & 0 & -2 & \infty \\ 1 & \infty & 0 & \infty \\ \infty & 5 & 3 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 1 & -1 & \\ & & & \\ & & & \\ & & & \end{pmatrix}$$

min{0+∞,3-2, ∞+0,-4+3}

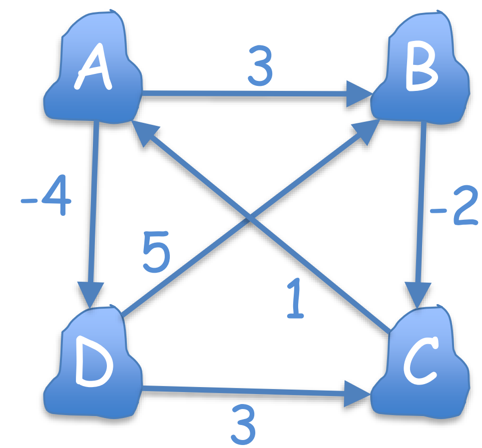
APSP

- The idea is to extend $P^{(1)}$ using the weight matrix W

$P^{(1)}$ = W
for m = 2 to n-1

$$P^{(m)} = Extend(P^{(m-1)}, W, n)$$

return $P^{(n-1)}$



$$P^{(2)} = Extend\left(P^{(1)}, W, n\right)$$
$$= P^{(1)} * W$$

$$P^{(2)} = \begin{pmatrix} 0 & 3 & \infty & -4 \\ \infty & 0 & -2 & \infty \\ 1 & \infty & 0 & \infty \\ \infty & 5 & 3 & 0 \end{pmatrix} \begin{pmatrix} 0 & 3 & \infty & -4 \\ \infty & 0 & -2 & \infty \\ 1 & \infty & 0 & \infty \\ \infty & 5 & 3 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 1 & -1 & -4 \\ -1 & 0 & -2 & \infty \\ 1 & 4 & 0 & -3 \\ 4 & 5 & 3 & 0 \end{pmatrix}$$

Floyd-Warshall

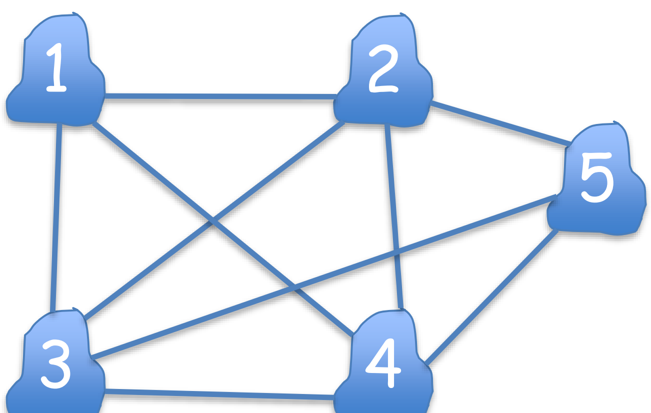
- For a given graph G with n vertices, all vertices are labaled as {1, 2, 3,…, n}

Floyd-Warshall

- For a given graph G with n vertices, all vertices are labaled as {1, 2, 3,…, n}
- For a path $\mathrm{p} = \langle v_1, v_2, \ldots, v_m \rangle$, $\{v_2, \ldots, v_{m-1}\}$ are called intermediate vertices
- Among all the paths from u to v, we consider only the ones in which all intermediate vertices are drawn from a subset {1, 2, … , k}

Floyd-Warshall

- For a given graph G with n vertices, all vertices are labaled as {1, 2, 3,…, n}
- For a path $\mathrm{p} = \langle v_1, v_2, \dots, v_m \rangle$, $\{v_2, \dots, v_{m-1}\}$ are called intermediate vertices
- Among all the paths from u to v, we consider only the ones in which all intermediate vertices are drawn from a subset {1, 2, … , k}



$1 \rightsquigarrow 5$

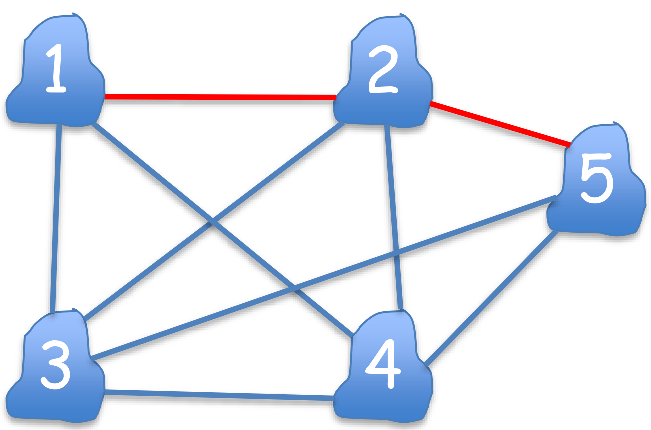
intermediate vertices are drawn from {1, 2, 3}

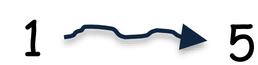
Floyd-Warshall

- For a given graph G with n vertices, all vertices are labaled as {1, 2, 3,…, n}

- For a path $\mathrm{p} = \langle v_1, v_2, \dots, v_m \rangle$, $\{v_2, \dots, v_{m-1}\}$ are called intermediate vertices

- Among all the paths from u to v, we consider only the ones in which all intermediate vertices are drawn from a subset {1, 2, … , k}

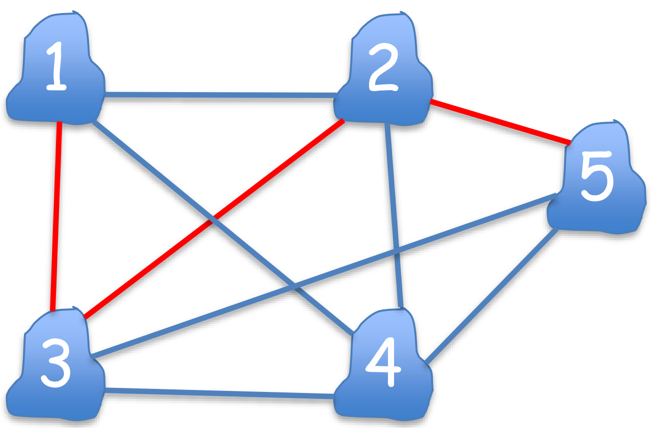


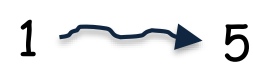


intermediate vertices are drawn from {1, 2, 3}

Floyd-Warshall

- For a given graph G with n vertices, all vertices are labaled as {1, 2, 3,…, n}
- For a path $\mathrm{p} = \langle v_1, v_2, \dots, v_m \rangle$, $\{v_2, \dots, v_{m-1}\}$ are called intermediate vertices
- Among all the paths from u to v, we consider only the ones in which all intermediate vertices are drawn from a subset {1, 2, … , k}





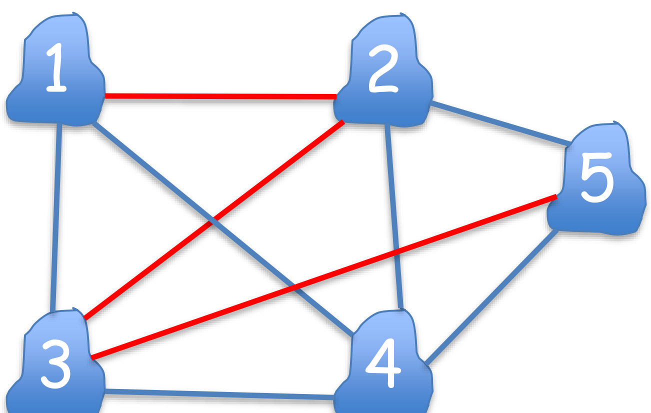
intermediate vertices are drawn from {1, 2, 3}

Floyd-Warshall

- For a given graph G with n vertices, all vertices are labaled as {1, 2, 3,…, n}

- For a path $\mathrm{p} = \langle v_1, v_2, \ldots, v_m \rangle$, $\{v_2, \ldots, v_{m-1}\}$ are called intermediate vertices

- Among all the paths from u to v, we consider only the ones in which all intermediate vertices are drawn from a subset {1, 2, … , k}

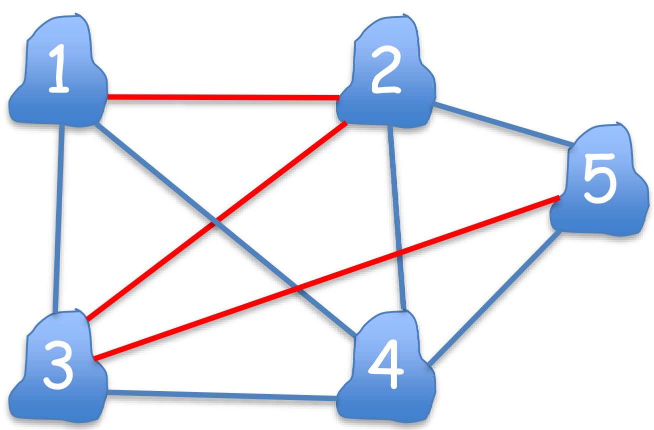


$1 \rightsquigarrow 5$

intermediate vertices are drawn from {1, 2, 3}

Floyd-Warshall

- For a given graph G with n vertices, all vertices are labaled as {1, 2, 3,…, n}
- For a path $\mathrm{p} = \langle v_1, v_2, \dots, v_m \rangle$, $\{v_2, \dots, v_{m-1}\}$ are called intermediate vertices
- Among all the paths from u to v, we consider only the ones in which all intermediate vertices are drawn from a subset {1, 2, … , k}



$1 \rightsquigarrow 5$

intermediate vertices are drawn from {1, 2, 3}

negative-weight cycle not allowed

Floyd-Warshall

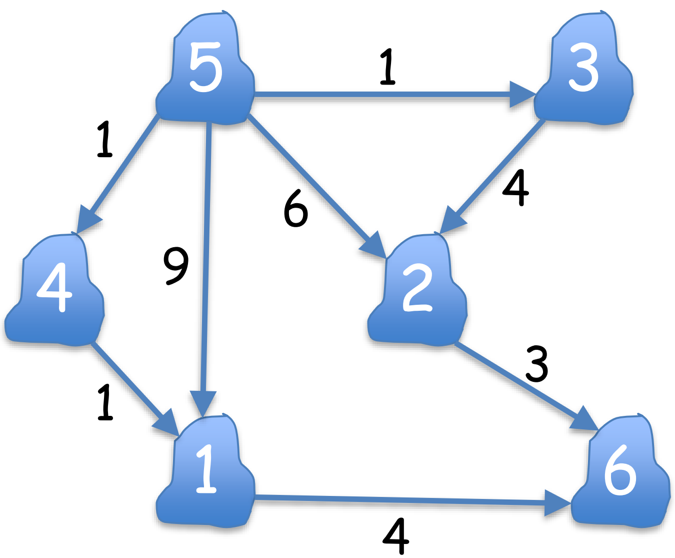
- define a subproblem

$d_{ij}^{(k)}$: the weight of a shortest path from vertex i to vertex j such that all intermediate vertices drawn from {1, 2, … , k}

Floyd-Warshall

- define a subproblem

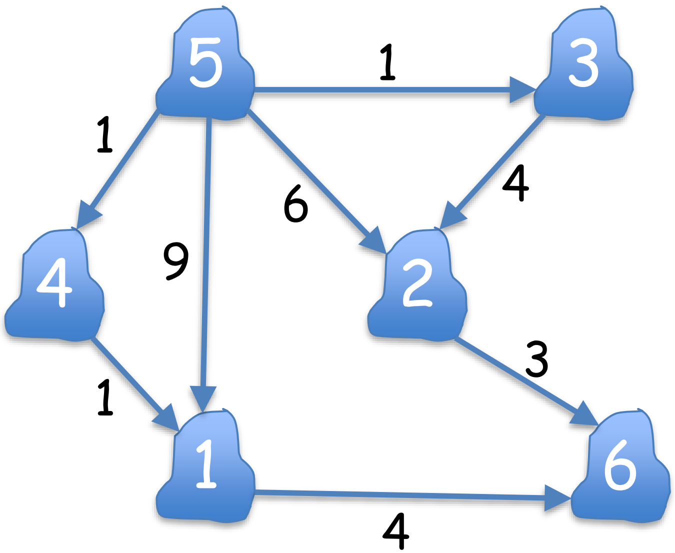
$d_{ij}^{(k)}$: the weight of a shortest path from vertex i to vertex j such that all intermediate vertices drawn from {1, 2, … , k}



Floyd-Warshall

- define a subproblem

$d_{ij}^{(k)}$: the weight of a shortest path from vertex i to vertex j such that all intermediate vertices drawn from {1, 2, … , k}

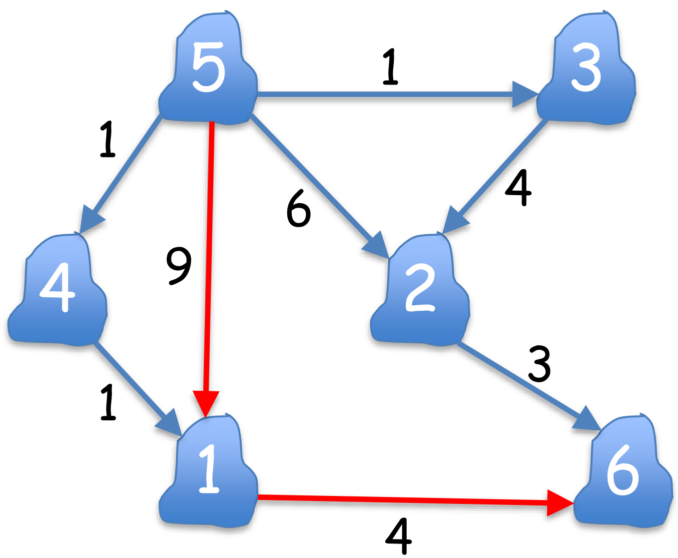


$d_{56}^{(0)} = \infty$

Floyd-Warshall

- define a subproblem

$d_{ij}^{(k)}$: the weight of a shortest path from vertex i to vertex j such that all intermediate vertices drawn from {1, 2, … , k}

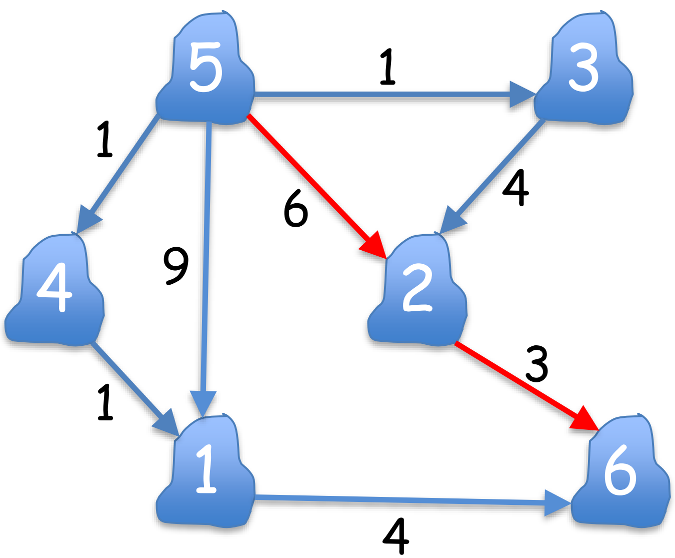


$$d_{56}^{(0)} = \infty$$
$$d_{56}^{(1)} = 13$$

Floyd-Warshall

- define a subproblem

$d_{ij}^{(k)}$: the weight of a shortest path from vertex i to vertex j such that all intermediate vertices drawn from {1, 2, … , k}

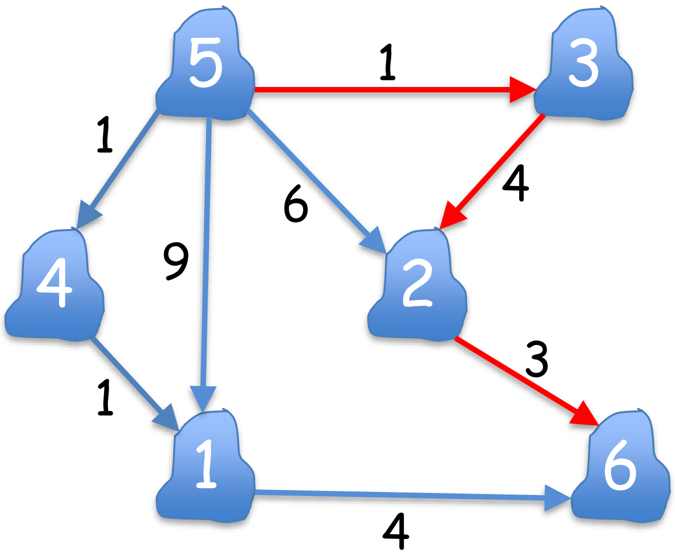


$$d_{56}^{(0)} = \infty$$
$$d_{56}^{(1)} = 13$$
$$d_{56}^{(2)} = 9$$

Floyd-Warshall

- define a subproblem

$d_{ij}^{(k)}$: the weight of a shortest path from vertex i to vertex j such that all intermediate vertices drawn from {1, 2, … , k}

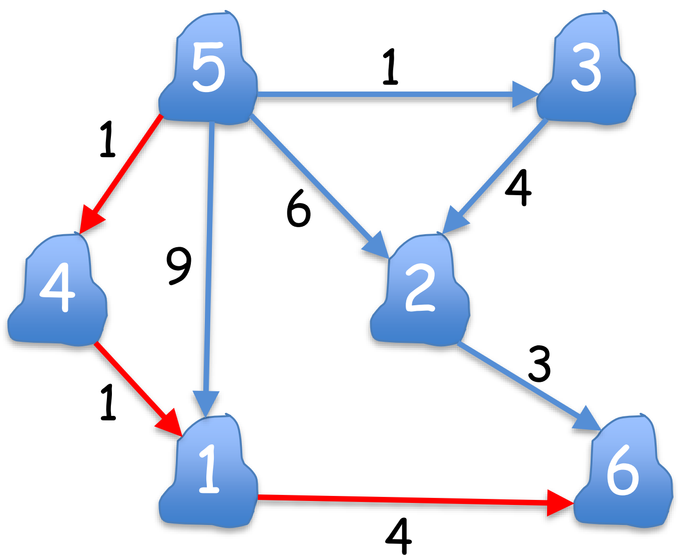


$$d_{56}^{(0)} = \infty$$
$$d_{56}^{(1)} = 13$$
$$d_{56}^{(2)} = 9$$
$$d_{56}^{(3)} = 8$$

Floyd-Warshall

- define a subproblem

$d_{ij}^{(k)}$: the weight of a shortest path from vertex i to vertex j such that all intermediate vertices drawn from {1, 2, … , k}



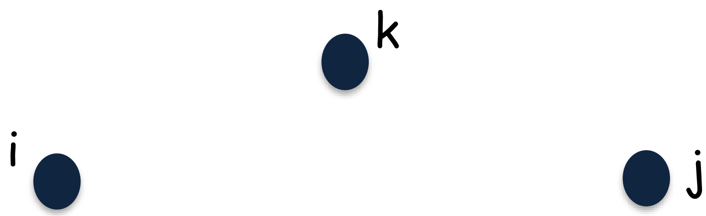
$d_{56}^{(0)} = \infty$
$d_{56}^{(1)} = 13$
$d_{56}^{(2)} = 9$
$d_{56}^{(3)} = 8$
$d_{56}^{(4)} = 6$

Floyd-Warshall

- define a subproblem

$d_{ij}^{(k)}$: the weight of a shortest path from vertex i to vertex j
such that all intermediate vertices drawn from {1, 2, … , k}

- construct recurrence relation



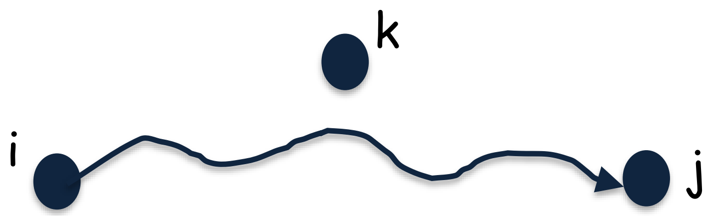
Two Cases

Floyd-Warshall

- define a subproblem

$d_{ij}^{(k)}$: the weight of a shortest path from vertex i to vertex j
such that all intermediate vertices drawn from {1, 2, … , k}

- construct recurrence relation



Two Cases

1) $d_{ij}^{(k)} = d_{ij}^{(k-1)}$

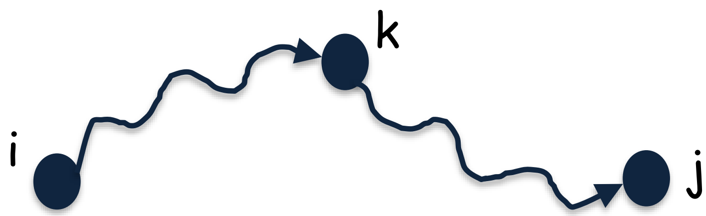
- k is not intermediate vertex in the path
(the intermediate vertices are drawn
from {1,2,…,k-1})

Floyd-Warshall

- define a subproblem

$d_{ij}^{(k)}$: the weight of a shortest path from vertex i to vertex j
such that all intermediate vertices drawn from {1, 2, … , k}

- construct recurrence relation



- k is an intermediate vertex in the path
(the intermediate vertices in the smaller paths are drawn from {1,2,…,k-1})

Two Cases

1) $d_{ij}^{(k)} = d_{ij}^{(k-1)}$

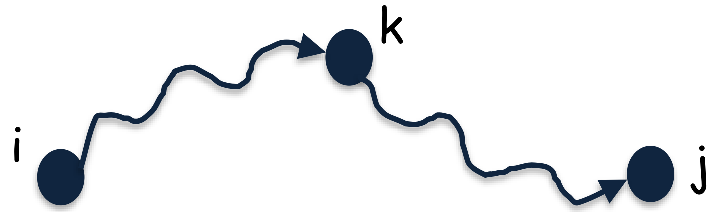
2) $d_{ij}^{(k)} = d_{ik}^{(k-1)} + d_{kj}^{(k-1)}$

Floyd-Warshall

- define a subproblem

$d_{ij}^{(k)}$: the weight of a shortest path from vertex i to vertex j such that all intermediate vertices drawn from {1, 2, … , k}

- construct recurrence relation



- k is an intermediate vertex in the path (the intermediate vertices in the smaller paths are drawn from {1,2,…,k-1})

Two Cases

1) $d_{ij}^{(k)} = d_{ij}^{(k-1)}$

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

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

Floyd-Warshall

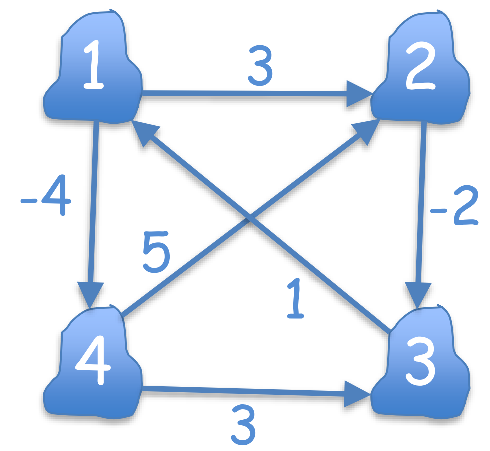
Floyd-Warshall

$D^{(0)}$=W
for k = 1 to n
    let $D^{(k)}$ be a new $n \times n$ matrix
    for i = 1 to n
        for j = 1 to n
            $d_{ij}^{(k)} = \min\left\{d_{ij}^{(k-1)}, d_{ik}^{(k-1)} + d_{kj}^{(k-1)}\right\}$

Floyd-Warshall

Floyd-Warshall

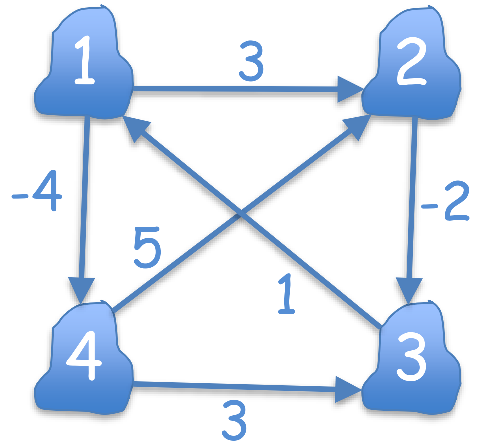
$D^{(0)}$=W
for k = 1 to n
  let $D^{(k)}$ be a new $n \times n$ matrix
  for i = 1 to n
    for j = 1 to n
      $d_{ij}^{(k)} = \min\left\{d_{ij}^{(k-1)}, d_{ik}^{(k-1)} + d_{kj}^{(k-1)}\right\}$



Floyd-Warshall

Floyd-Warshall

$D^{(0)}$=W
for k = 1 to n
    let $D^{(k)}$ be a new $n \times n$ matrix
    for i = 1 to n
        for j = 1 to n
            $d_{ij}^{(k)} = \min\left\{d_{ij}^{(k-1)}, d_{ik}^{(k-1)} + d_{kj}^{(k-1)}\right\}$

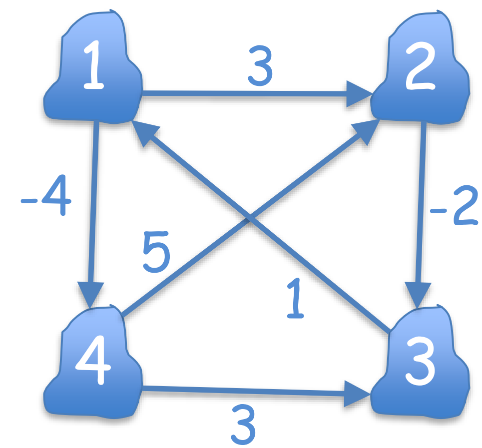


$$D^{(0)} = \begin{pmatrix} 0 & 3 & \infty & -4 \\ \infty & 0 & -2 & \infty \\ 1 & \infty & 0 & \infty \\ \infty & 5 & 3 & 0 \end{pmatrix}$$

Floyd-Warshall

Floyd-Warshall

$D^{(0)}$=W
for k = 1 to n
    let $D^{(k)}$ be a new $n \times n$ matrix
    for i = 1 to n
        for j = 1 to n
            $d_{ij}^{(k)} = \min\left\{d_{ij}^{(k-1)}, d_{ik}^{(k-1)} + d_{kj}^{(k-1)}\right\}$

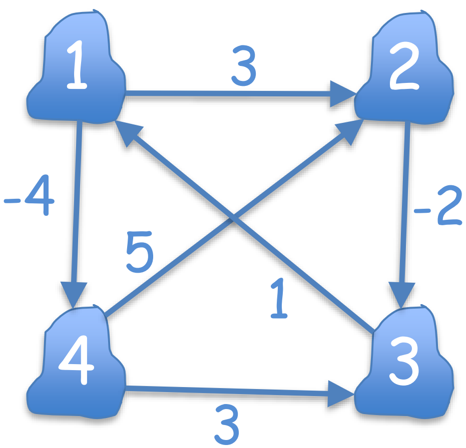


$$D^{(0)} = \begin{pmatrix} 0 & 3 & \infty & -4 \\ \infty & 0 & -2 & \infty \\ 1 & \infty & 0 & \infty \\ \infty & 5 & 3 & 0 \end{pmatrix} \longrightarrow D^{(1)} = \begin{pmatrix} 0 & 3 & \infty & -4 \\ \infty & 0 & -2 & \infty \\ 1 & & & \\ & & & \end{pmatrix}$$

Floyd-Warshall

Floyd-Warshall

$D^{(0)}$=W
for k = 1 to n
  let $D^{(k)}$ be a new $n \times n$ matrix
  for i = 1 to n
    for j = 1 to n
      $d_{ij}^{(k)} = \min\left\{d_{ij}^{(k-1)}, d_{ik}^{(k-1)} + d_{kj}^{(k-1)}\right\}$



$$D^{(0)} = \begin{pmatrix} 0 & 3 & \infty & -4 \\ \infty & 0 & -2 & \infty \\ 1 & \infty & 0 & \infty \\ \infty & 5 & 3 & 0 \end{pmatrix}$$

$$D^{(1)} = \begin{pmatrix} 0 & 3 & \infty & -4 \\ \infty & 0 & -2 & \infty \\ 1 & & & \\ & & & \end{pmatrix}$$

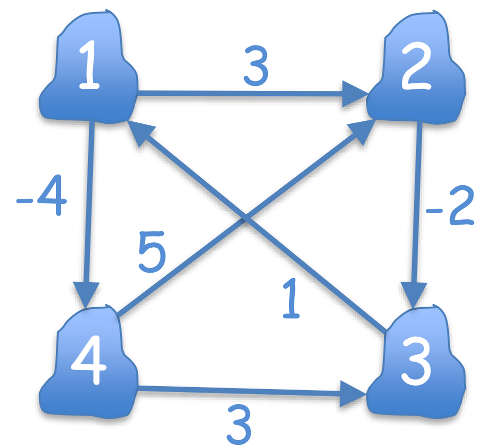
$$d_{32}^{(1)} = \min\left\{d_{32}^{(0)}, d_{31}^{(0)} + d_{12}^{(0)}\right\}$$

Floyd-Warshall

Floyd-Warshall

$D^{(0)}$=W
for k = 1 to n
    let $D^{(k)}$ be a new $n \times n$ matrix
    for i = 1 to n
        for j = 1 to n

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



$$D^{(0)} = \begin{pmatrix} 0 & 3 & \infty & -4 \\ \infty & 0 & -2 & \infty \\ 1 & \infty & 0 & \infty \\ \infty & 5 & 3 & 0 \end{pmatrix}$$

$$D^{(1)} = \begin{pmatrix} 0 & 3 & \infty & -4 \\ \infty & 0 & -2 & \infty \\ 1 & 4 & & \\ & & & \end{pmatrix}$$

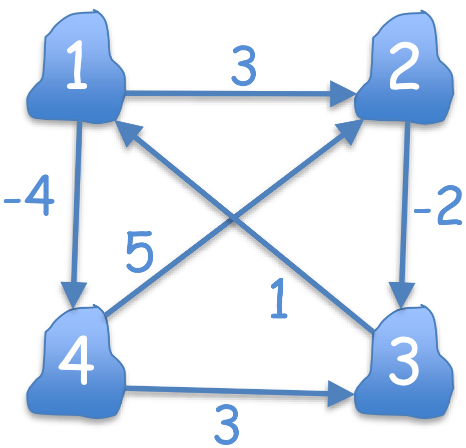
$$d_{32}^{(1)} = \min\left\{d_{32}^{(0)}, d_{31}^{(0)} + d_{12}^{(0)}\right\}$$

$$d_{32}^{(1)} = \min\{\infty\ , 1 + 3\} = 4$$

Floyd-Warshall

<u>Floyd-Warshall</u>

$D^{(0)}$=W
for k = 1 to n
    let $D^{(k)}$ be a new $n \times n$ matrix
    for i = 1 to n
        for j = 1 to n
            $d_{ij}^{(k)} = \min\left\{d_{ij}^{(k-1)}, d_{ik}^{(k-1)} + d_{kj}^{(k-1)}\right\}$



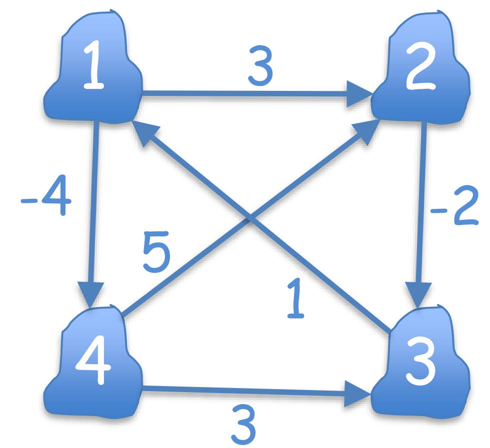
$$D^{(0)} = \begin{pmatrix} 0 & 3 & \infty & -4 \\ \infty & 0 & -2 & \infty \\ 1 & \infty & 0 & \infty \\ \infty & 5 & 3 & 0 \end{pmatrix}$$

$$D^{(1)} = \begin{pmatrix} 0 & 3 & \infty & -4 \\ \infty & 0 & -2 & \infty \\ 1 & 4 & 0 & \\ & & & \end{pmatrix}$$

Floyd-Warshall

Floyd-Warshall

$D^{(0)}$=W
for k = 1 to n
    let $D^{(k)}$ be a new $n \times n$ matrix
    for i = 1 to n
        for j = 1 to n
            $d_{ij}^{(k)} = \min\left\{d_{ij}^{(k-1)}, d_{ik}^{(k-1)} + d_{kj}^{(k-1)}\right\}$



$$D^{(0)} = \begin{pmatrix} 0 & 3 & \infty & -4 \\ \infty & 0 & -2 & \infty \\ 1 & \infty & 0 & \infty \\ \infty & 5 & 3 & 0 \end{pmatrix}$$

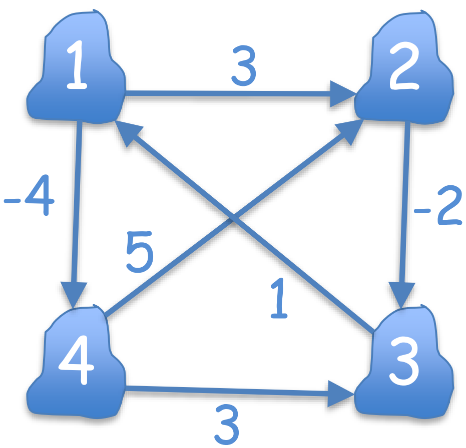
$$D^{(1)} = \begin{pmatrix} 0 & 3 & \infty & -4 \\ \infty & 0 & -2 & \infty \\ 1 & 4 & 0 & \\ & & & \end{pmatrix}$$

$$d_{34}^{(1)} = \min\left\{d_{34}^{(0)}, d_{31}^{(0)} + d_{14}^{(0)}\right\}$$

Floyd-Warshall

Floyd-Warshall

$D^{(0)}$=W
for k = 1 to n
    let $D^{(k)}$ be a new $n \times n$ matrix
    for i = 1 to n
        for j = 1 to n
            $d_{ij}^{(k)} = \min\left\{d_{ij}^{(k-1)}, d_{ik}^{(k-1)} + d_{kj}^{(k-1)}\right\}$

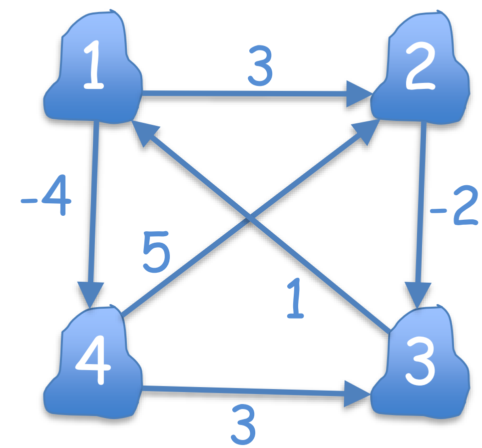


$$D^{(0)} = \begin{pmatrix} 0 & 3 & \infty & -4 \\ \infty & 0 & -2 & \infty \\ 1 & \infty & 0 & \infty \\ \infty & 5 & 3 & 0 \end{pmatrix}$$

$$D^{(1)} = \begin{pmatrix} 0 & 3 & \infty & -4 \\ \infty & 0 & -2 & \infty \\ 1 & 4 & 0 & -3 \\ & & & \end{pmatrix}$$

$$d_{34}^{(1)} = \min\left\{d_{34}^{(0)}, d_{31}^{(0)} + d_{14}^{(0)}\right\}$$
$$d_{34}^{(1)} = \min\{\infty\ , 1 - 4\} = -3$$

Floyd-Warshall



Floyd-Warshall

$D^{(0)}$=W
for k = 1 to n
    let $D^{(k)}$ be a new $n \times n$ matrix
    for i = 1 to n
        for j = 1 to n

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

$$D^{(0)} = \begin{pmatrix} 0 & 3 & \infty & -4 \\ \infty & 0 & -2 & \infty \\ 1 & \infty & 0 & \infty \\ \infty & 5 & 3 & 0 \end{pmatrix}$$

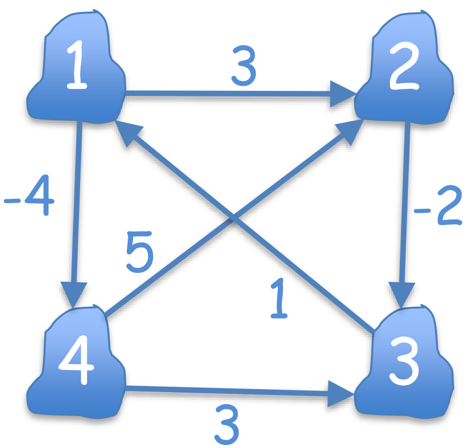
$$D^{(1)} = \begin{pmatrix} 0 & 3 & \infty & -4 \\ \infty & 0 & -2 & \infty \\ 1 & 4 & 0 & -3 \\ \infty & 5 & 3 & 0 \end{pmatrix}$$

Floyd-Warshall

<u>Floyd-Warshall</u>

$D^{(0)}$=W
for k = 1 to n
    let $D^{(k)}$ be a new $n \times n$ matrix
    for i = 1 to n
        for j = 1 to n

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



$$D^{(0)} = \begin{pmatrix} 0 & 3 & \infty & -4 \\ \infty & 0 & -2 & \infty \\ 1 & \infty & 0 & \infty \\ \infty & 5 & 3 & 0 \end{pmatrix}$$

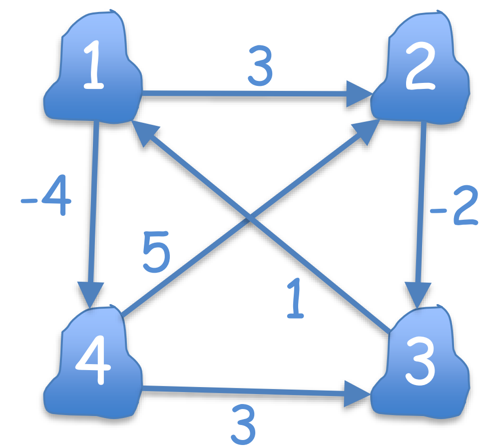
$$D^{(1)} = \begin{pmatrix} 0 & 3 & \infty & -4 \\ \infty & 0 & -2 & \infty \\ 1 & 4 & 0 & -3 \\ \infty & 5 & 3 & 0 \end{pmatrix}$$

$$D^{(2)} = \begin{pmatrix} 0 & 3 & 1 & -4 \\ \infty & 0 & -2 & \infty \\ 1 & 4 & 0 & -3 \\ \infty & 5 & 3 & 0 \end{pmatrix}$$

Floyd-Warshall

<u>Floyd-Warshall</u>

$D^{(0)}$=W
for k = 1 to n
    let $D^{(k)}$ be a new $n \times n$ matrix
    for i = 1 to n
        for j = 1 to n
            $d_{ij}^{(k)} = \min\left\{d_{ij}^{(k-1)}, d_{ik}^{(k-1)} + d_{kj}^{(k-1)}\right\}$



$$D^{(0)} = \begin{pmatrix} 0 & 3 & \infty & -4 \\ \infty & 0 & -2 & \infty \\ 1 & \infty & 0 & \infty \\ \infty & 5 & 3 & 0 \end{pmatrix}$$

$$D^{(1)} = \begin{pmatrix} 0 & 3 & \infty & -4 \\ \infty & 0 & -2 & \infty \\ 1 & 4 & 0 & -3 \\ \infty & 5 & 3 & 0 \end{pmatrix}$$

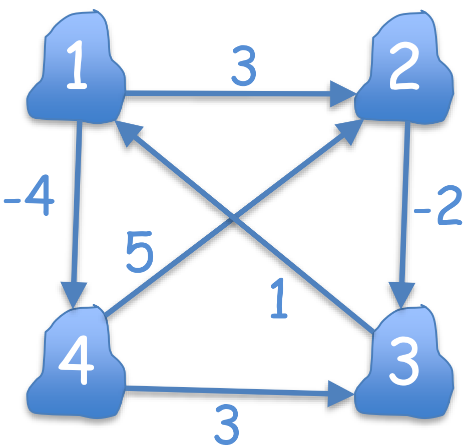
$$D^{(2)} = \begin{pmatrix} 0 & 3 & 1 & -4 \\ \infty & 0 & -2 & \infty \\ 1 & 4 & 0 & -3 \\ \infty & 5 & 3 & 0 \end{pmatrix}$$

$$D^{(3)} = \begin{pmatrix} 0 & 3 & 1 & -4 \\ -1 & 0 & -2 & -5 \\ 1 & 4 & 0 & -3 \\ 4 & 5 & 3 & 0 \end{pmatrix}$$

Floyd-Warshall

Floyd-Warshall

$D^{(0)}$=W
for k = 1 to n
  let $D^{(k)}$ be a new $n \times n$ matrix
  for i = 1 to n
    for j = 1 to n
      $d_{ij}^{(k)} = \min\left\{d_{ij}^{(k-1)}, d_{ik}^{(k-1)} + d_{kj}^{(k-1)}\right\}$



$$D^{(0)} = \begin{pmatrix} 0 & 3 & \infty & -4 \\ \infty & 0 & -2 & \infty \\ 1 & \infty & 0 & \infty \\ \infty & 5 & 3 & 0 \end{pmatrix}$$

$$D^{(1)} = \begin{pmatrix} 0 & 3 & \infty & -4 \\ \infty & 0 & -2 & \infty \\ 1 & 4 & 0 & -3 \\ \infty & 5 & 3 & 0 \end{pmatrix}$$

$$D^{(2)} = \begin{pmatrix} 0 & 3 & 1 & -4 \\ \infty & 0 & -2 & \infty \\ 1 & 4 & 0 & -3 \\ \infty & 5 & 3 & 0 \end{pmatrix}$$

$$D^{(3)} = \begin{pmatrix} 0 & 3 & 1 & -4 \\ -1 & 0 & -2 & -5 \\ 1 & 4 & 0 & -3 \\ 4 & 5 & 3 & 0 \end{pmatrix}$$

$$D^{(4)} = \begin{pmatrix} 0 & 1 & -1 & -4 \\ -1 & 0 & -2 & -5 \\ 1 & 2 & 0 & -3 \\ 4 & 5 & 3 & 0 \end{pmatrix}$$