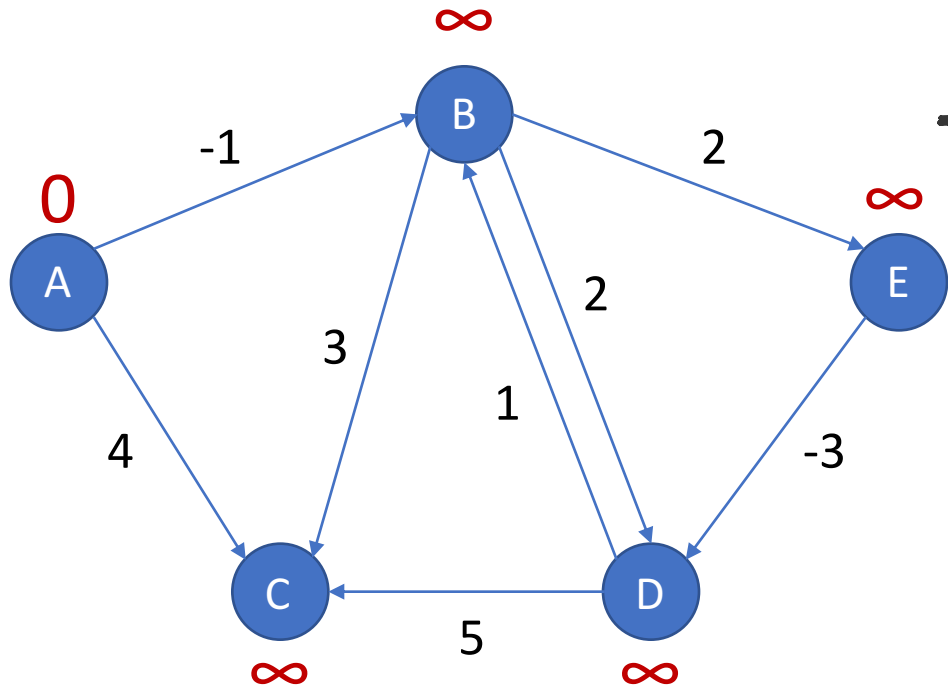


Bellman-Ford Algorithm Example

Initialization:



A	B	C	D	E
0	∞	∞	∞	∞

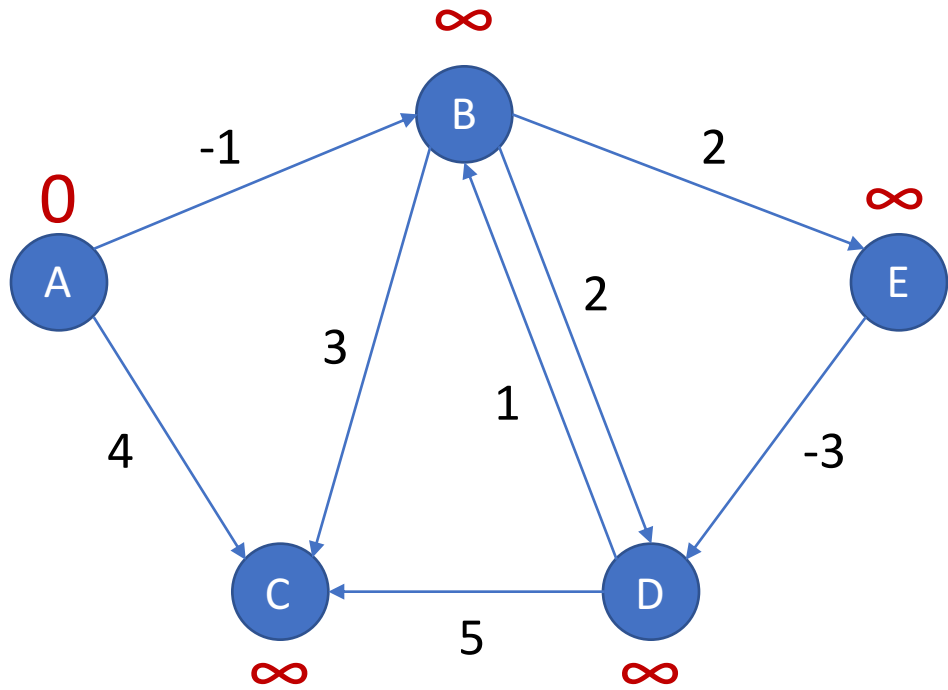
u

```
for each u in V  
  d(u,0) = infity  
d(s,0) = 0
```

```
for k = 1 to n-1  
  for each v in V  
    d(v,k) = d(v, k-1)  
    for each edge in incoming(V)  
      d(v,k) = min{d(v, k-1),  
        d(u, k-1) + l(u,v)}
```

```
for each v in V do  
  dist(s,v) = d(v, n-1)
```

k=1:



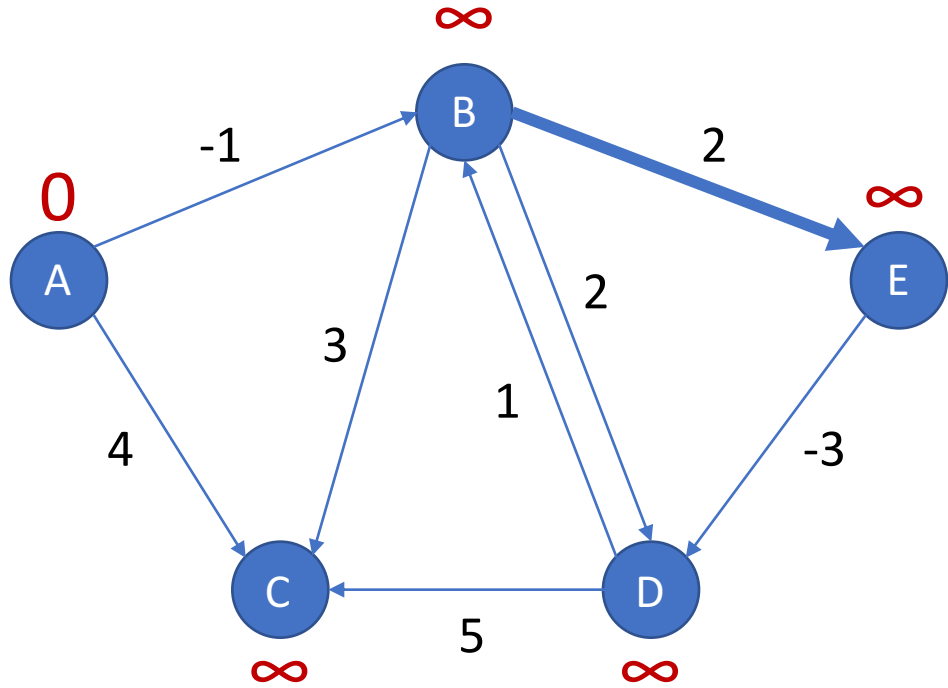
A	B	C	D	E
0	∞	∞	∞	∞
0	∞	∞	∞	∞

```
for each u in V
  d(u,0) = infty
d(s,0) = 0
```

```
for k = 1 to n-1
  for each v in V
    d(v,k) = d(v, k-1)
    for each edge in incoming(V)
      d(v,k) = min{d(v,k-1), d(u,k-1) + l(u,v)}
```

```
[ for each v in V do
  dist(s,v) = d(v, n-1)
```

k=1:



$d(v, \cdot)$

	A	B	C	D	E
0	0	∞	∞	∞	∞
1	0	∞	∞	∞	∞

$d(E, 1) = \min \begin{cases} d(E, 0) = \infty \\ d(B, 0) + 2 = \infty \end{cases}$

```

for each u in V
  d(u, 0) = infty
d(s, 0) = 0

```

```

for k = 1 to n-1
  for each v in V
    d(v, k) = d(v, k-1)
    for each edge in incoming(V)
      d(v, k) = min{d(v, k),
                    d(u, k-1) + l(u, v)}

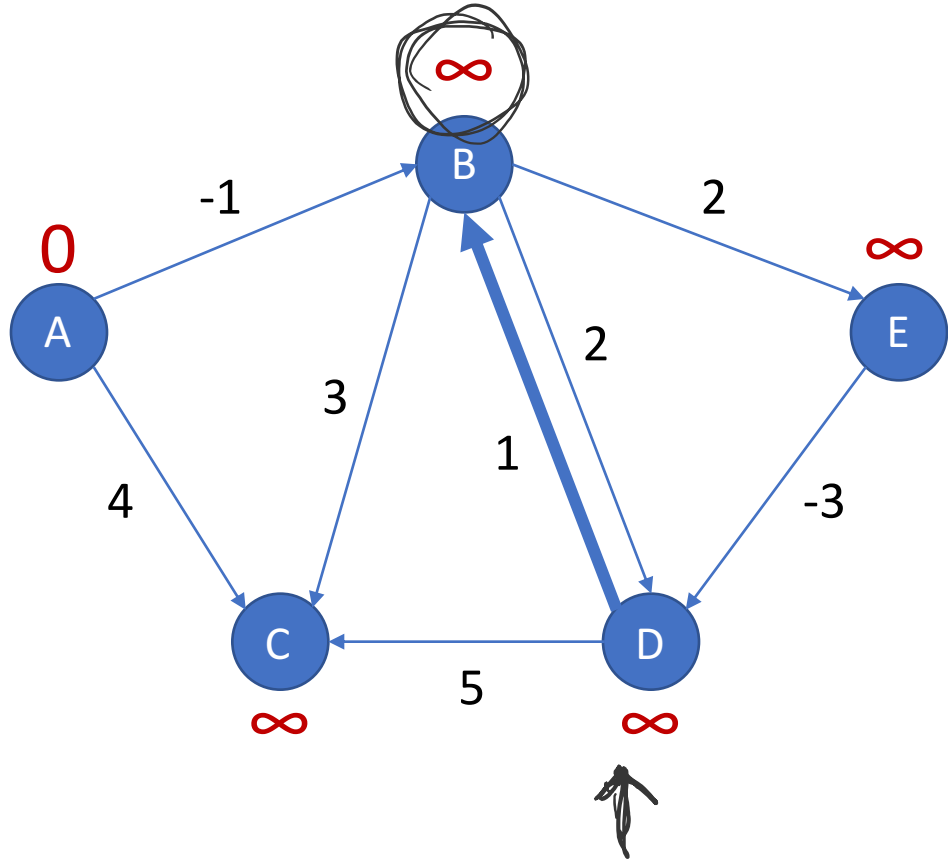
```

```

for each v in V do
  dist(s, v) = d(v, n-1)

```

k=1:



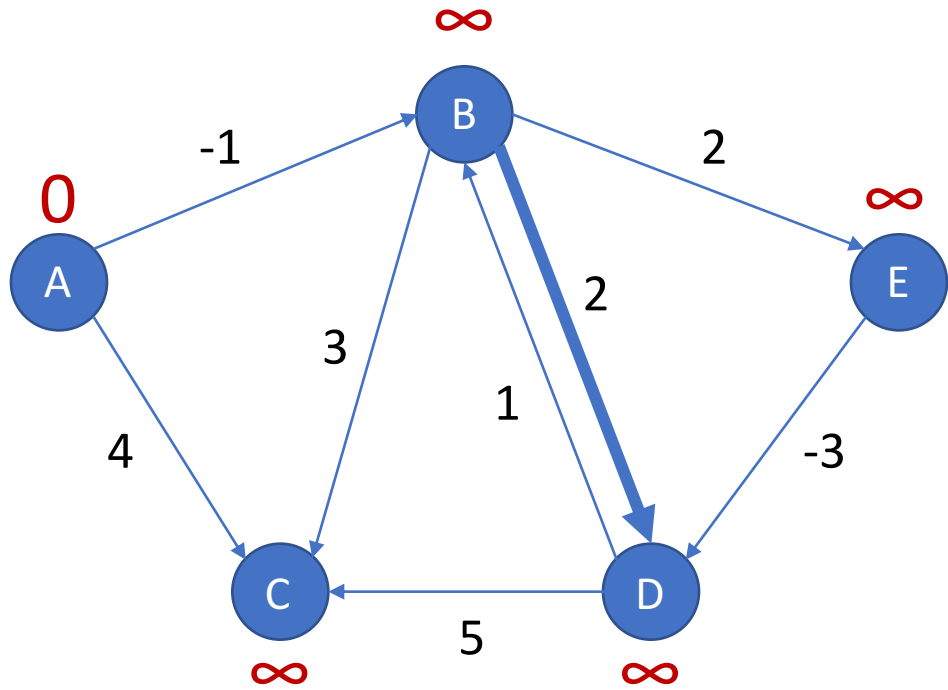
A	B	C	D	E
0	∞	∞	∞	∞
0	∞	∞	∞	∞

```
for each u in V
  d(u,0) = infty
d(s,0) = 0

for k = 1 to n-1
  for each v in V
    d(v,k) = d(v, k-1)
    for each edge in incoming(V)
      d(v,k) = min{d(v,k),
                  d(u,k-1)+l(u,v)}

for each v in V do
  dist(s,v) = d(v, n-1)
```

k=1:



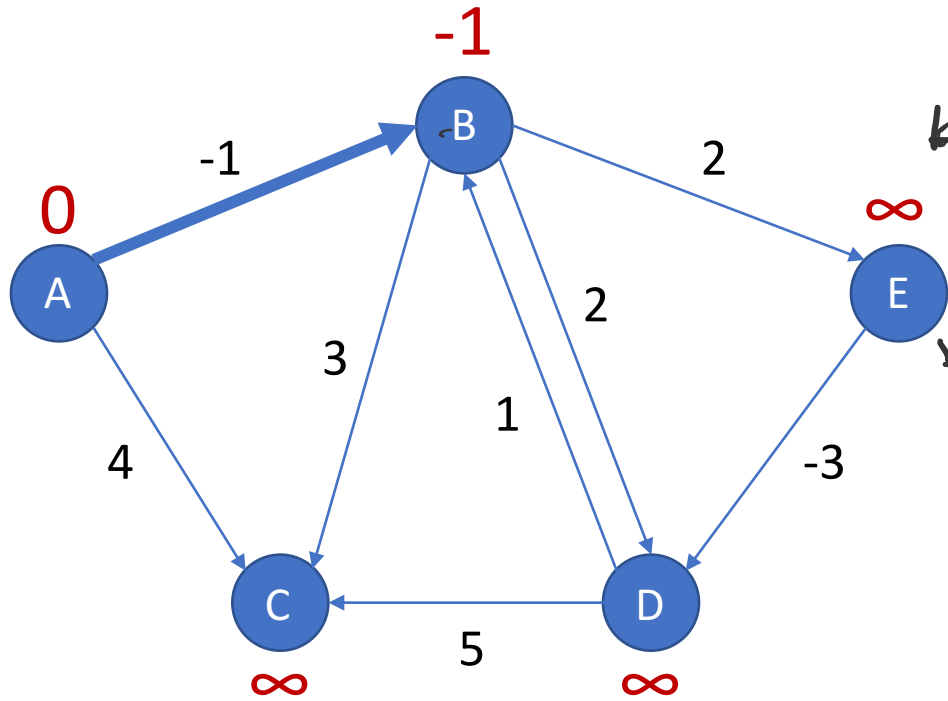
A	B	C	D	E
0	∞	∞	∞	∞
0	∞	∞	∞	∞

```
for each u in V
  d(u,0) = infty
d(s,0) = 0

for k = 1 to n-1
  for each v in V
    d(v,k) = d(v, k-1)
    for each edge in incoming(V)
      d(v,k) = min{d(v,k),
                  d(u,k-1)+l(u,v)}

for each v in V do
  dist(s,v) = d(v, n-1)
```

k=1:



	A	B	C	D	E
k=0	0	∞	∞	∞	∞
k=1	0	-1	∞	∞	∞

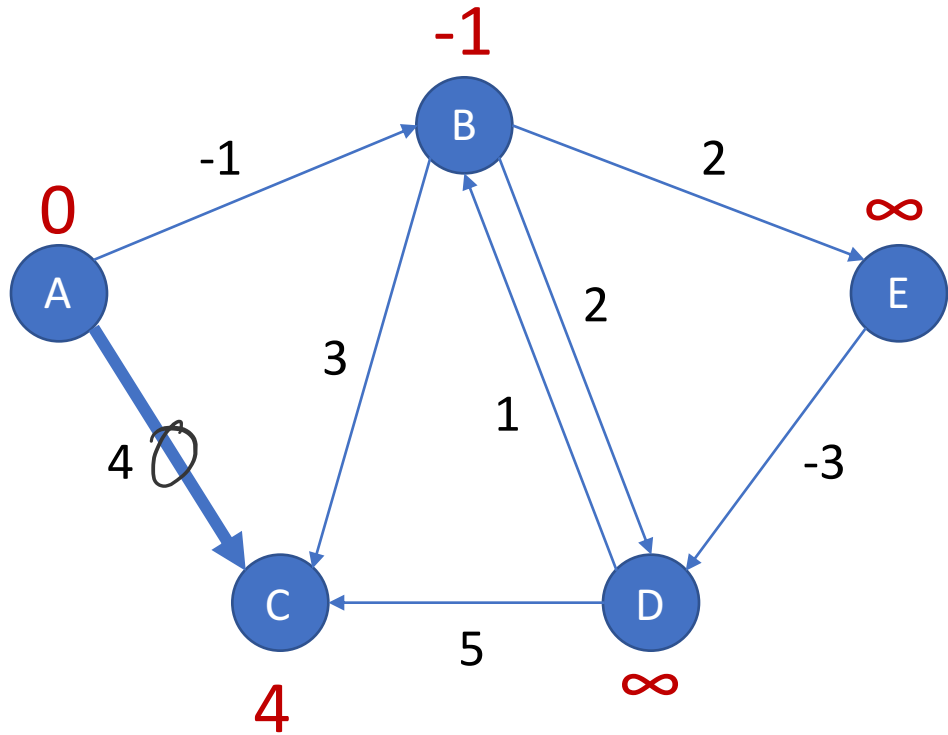
$$d(B,1) = \min \begin{cases} d(B,0) = \infty \\ d(A,0) + -1 \end{cases}$$

```
for each u in V
  d(u,0) = infty
d(s,0) = 0
```

```
for k = 1 to n-1
  for each v in V
    d(v,k) = d(v, k-1)
    for each edge in incoming(V)
      d(v,k) = min{d(v,k),
                  d(u,k-1)+l(u,v)}
```

```
for each v in V do
  dist(s,v) = d(v, n-1)
```

k=1:



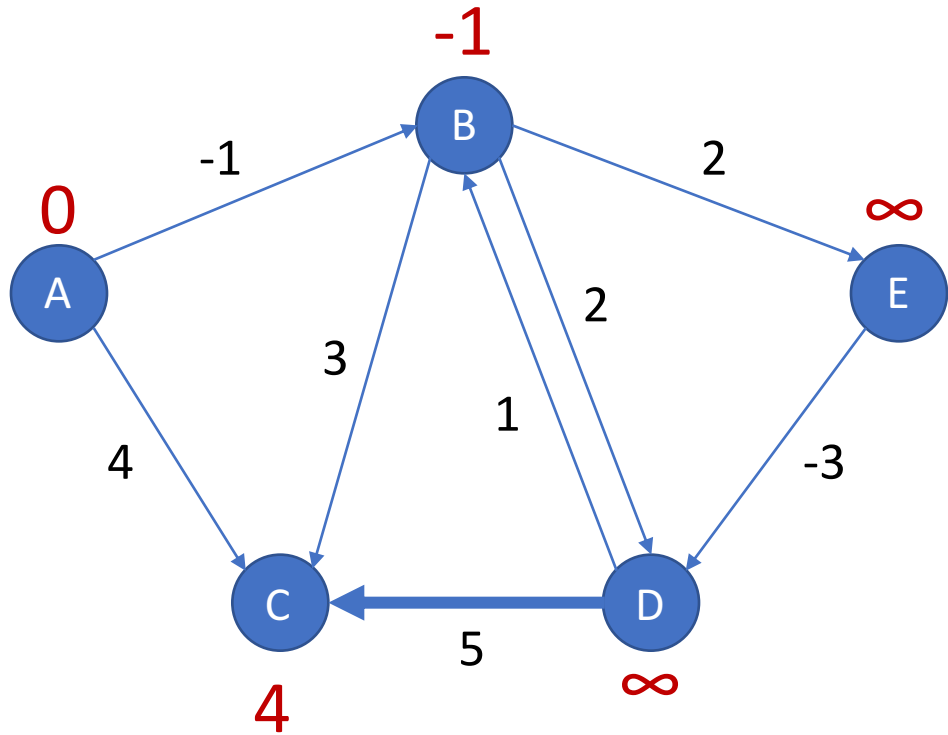
A	B	C	D	E
0	∞	∞	∞	∞
0	∞	∞	∞	∞
0	-1	∞	∞	∞
0	-1	4	∞	∞

```
for each u in V
  d(u,0) = infty
d(s,0) = 0

for k = 1 to n-1
  for each v in V
    d(v,k) = d(v, k-1)
    for each edge in incoming(V)
      d(v,k) = min{d(v,k),
                  d(u,k-1)+l(u,v)}

for each v in V do
  dist(s,v) = d(v, n-1)
```


k=1:



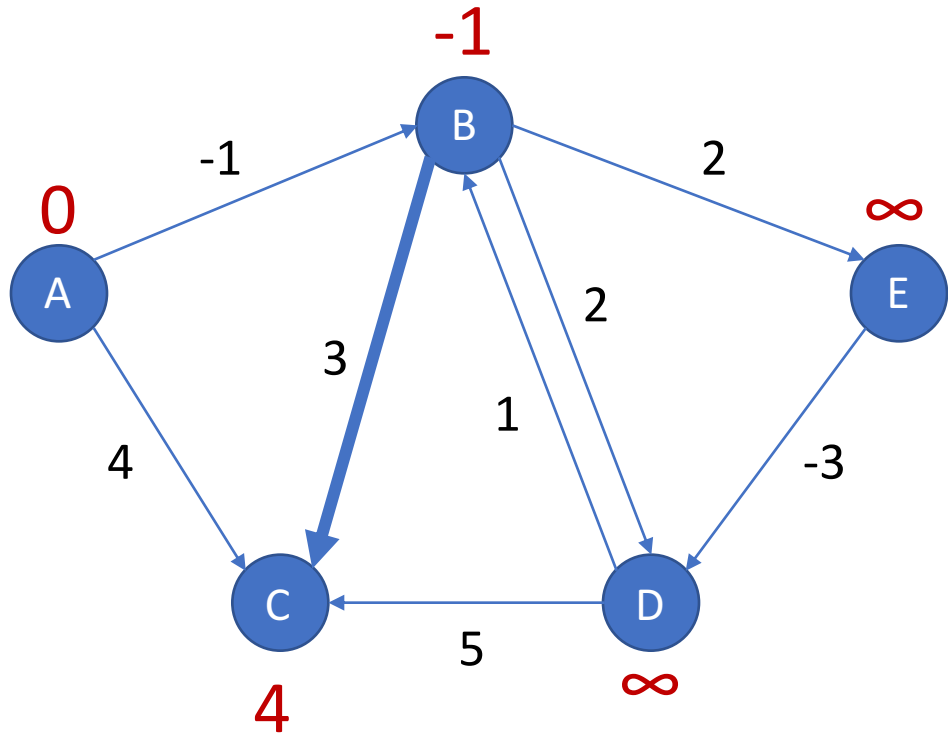
	A	B	C	D	E
A	0	∞	∞	∞	∞
B	0	∞	∞	∞	∞
C	0	-1	∞	∞	∞
D	0	-1	4	∞	∞

```
for each u in V
  d(u,0) = infty
d(s,0) = 0

for k = 1 to n-1
  for each v in V
    d(v,k) = d(v, k-1)
    for each edge in incoming(V)
      d(v,k) = min{d(v,k),
                  d(u,k-1)+l(u,v)}

for each v in V do
  dist(s,v) = d(v, n-1)
```

k=1:



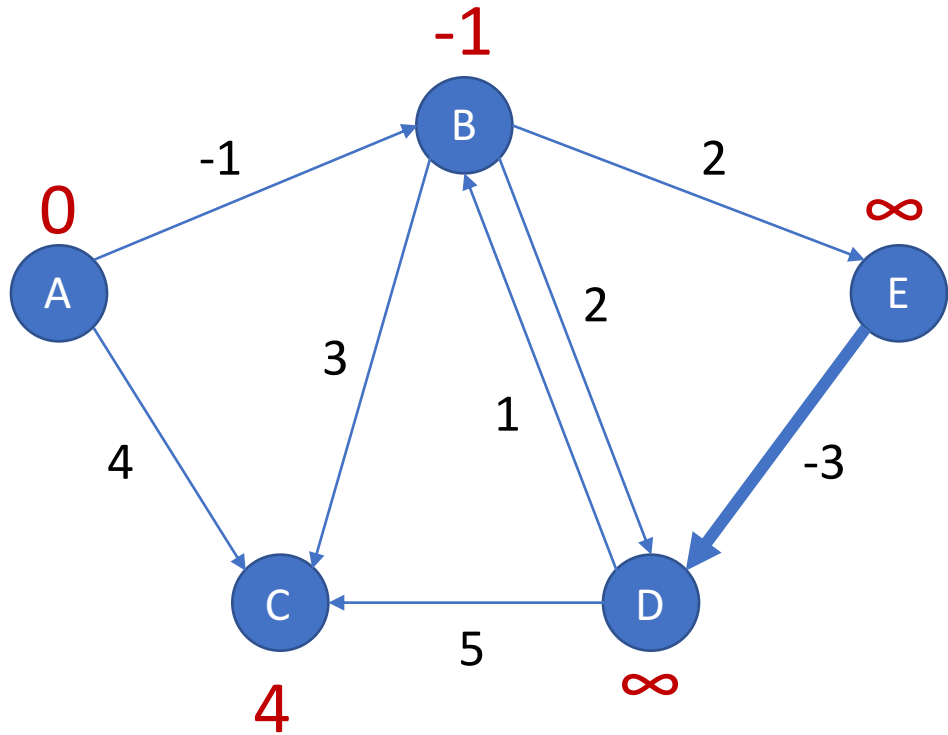
	A	B	C	D	E
A	0	∞	∞	∞	∞
B	0	∞	∞	∞	∞
C	0	-1	∞	∞	∞
D	0	-1	4	∞	∞

```
for each u in V
  d(u,0) = infty
d(s,0) = 0

for k = 1 to n-1
  for each v in V
    d(v,k) = d(v, k-1)
    for each edge in incoming(V)
      d(v,k) = min{d(v,k),
                  d(u,k-1)+l(u,v)}

for each v in V do
  dist(s,v) = d(v, n-1)
```

k=1:



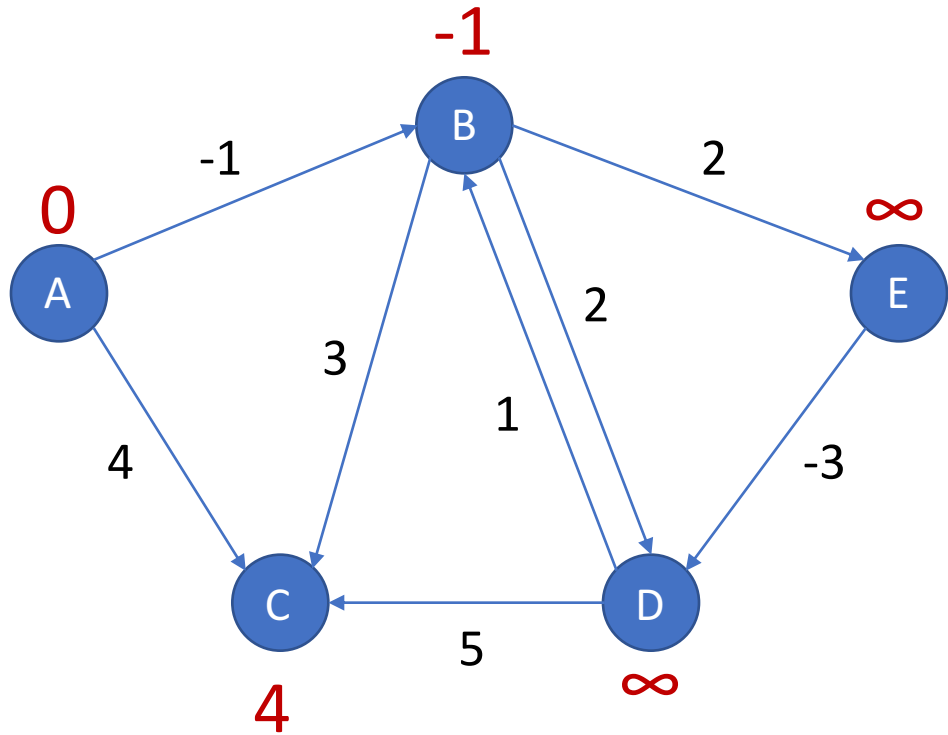
A	B	C	D	E
0	∞	∞	∞	∞
0	∞	∞	∞	∞
0	-1	∞	∞	∞
0	-1	4	∞	∞

```
for each u in V
  d(u,0) = infty
d(s,0) = 0

for k = 1 to n-1
  for each v in V
    d(v,k) = d(v, k-1)
    for each edge in incoming(V)
      d(v,k) = min{d(v,k),
                  d(u,k-1)+l(u,v)}

for each v in V do
  dist(s,v) = d(v, n-1)
```

k=1:



A	B	C	D	E
0	∞	∞	∞	∞
0	∞	∞	∞	∞
0	-1	∞	∞	∞
0	-1	4	∞	∞

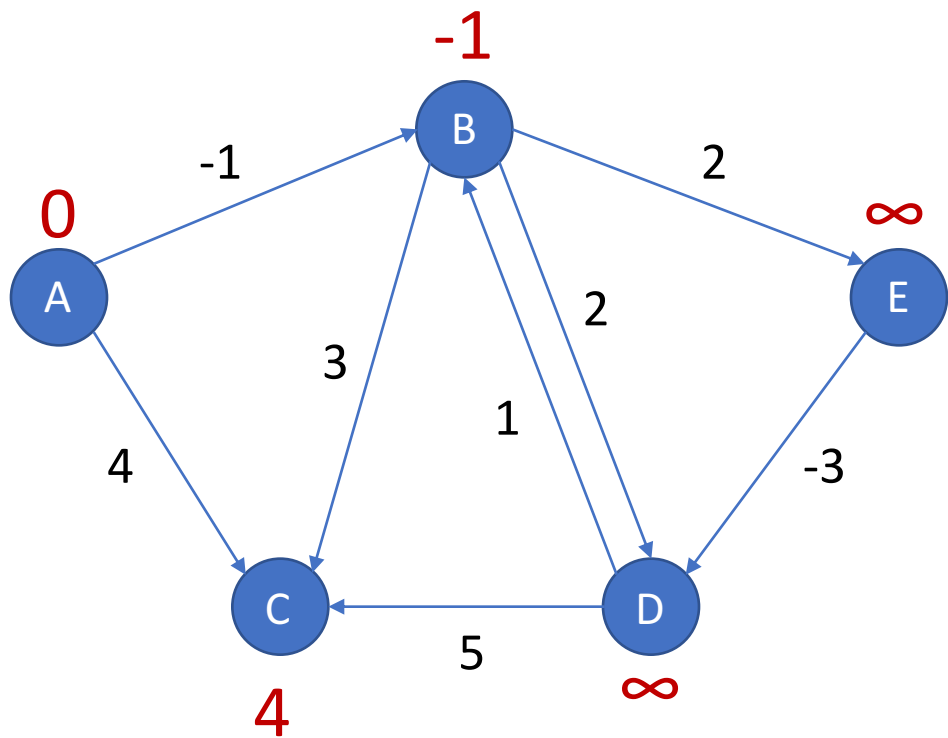
```
for each u in V
  d(u,0) = infty
d(s,0) = 0

for k = 1 to n-1
  for each v in V
    d(v,k) = d(v, k-1)
    for each edge in incoming(V)
      d(v,k) = min{d(v,k),
                  d(u,k-1)+l(u,v)}

for each v in V do
  dist(s,v) = d(v, n-1)
```

Distances assuming all paths lengths are at most 1 edge long.

k=2:



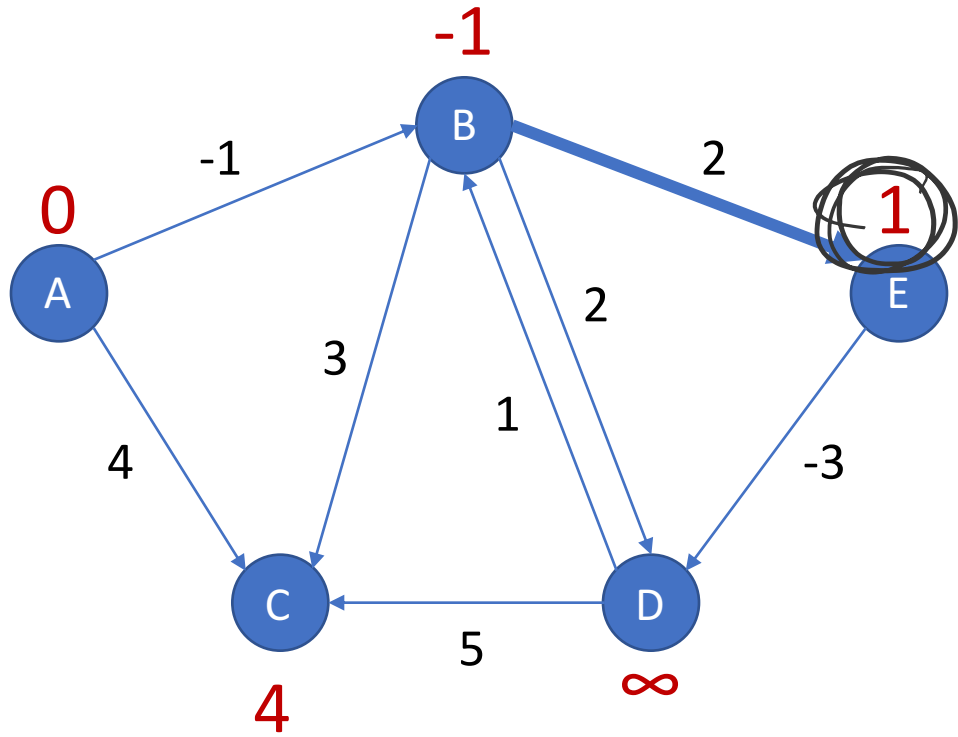
	A	B	C	D	E
A	0	∞	∞	∞	∞
B	0	∞	∞	∞	∞
C	0	-1	∞	∞	∞
D	0	-1	4	∞	∞
E	0	-1	4	∞	∞

```
for each u in V
  d(u,0) = infity
d(s,0) = 0

for k = 1 to n-1
  for each v in V
    d(v,k) = d(v, k-1)
    for each edge in incoming(V)
      d(v,k) = min{d(v,k),
                  d(u,k-1)+l(u,v)}

for each v in V do
  dist(s,v) = d(v, n-1)
```

k=2:



A	B	C	D	E
0	∞	∞	∞	∞
0	∞	∞	∞	∞
0	-1	∞	∞	∞
0	-1	4	∞	∞
0	-1	4	∞	∞
0	-1	4	∞	1

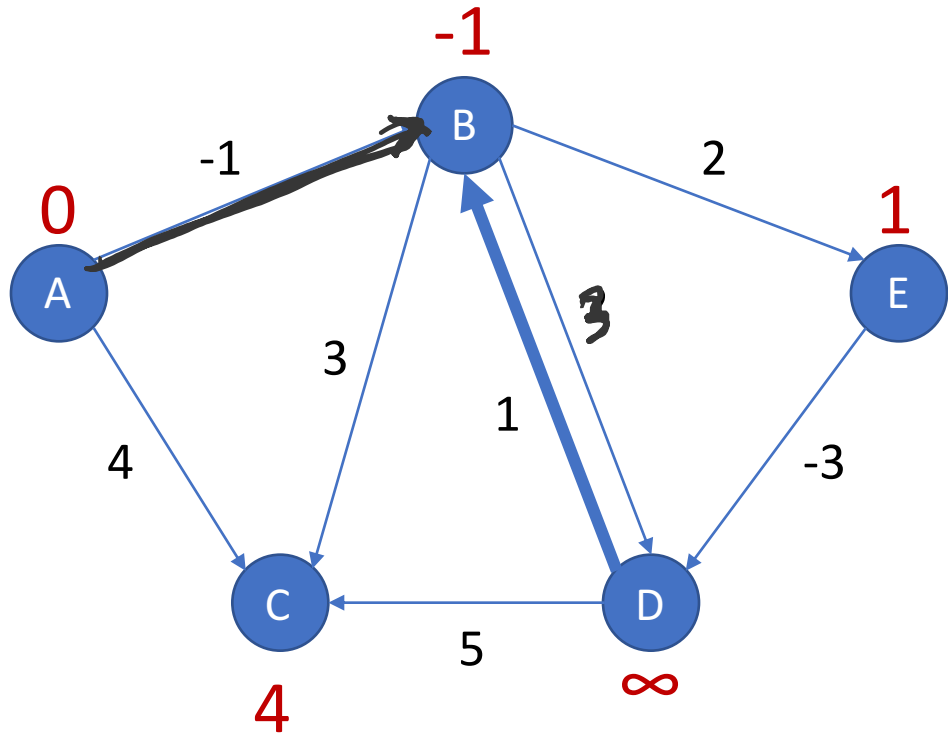
for each u in V
 $d(u,0) = \text{infty}$
 $d(s,0) = 0$

for k = 1 to n-1
 for each v in V
 $d(v,k) = d(v, k-1)$
 for each edge in incoming(V)
 $d(v,k) = \min\{d(v,k), d(u,k-1) + l(u,v)\}$

for each v in V do
 $\text{dist}(s,v) = d(v, n-1)$

$$d(E,2) = \min \begin{cases} d(E,0) = \infty \\ d(B,0) + 2 \\ -1 \end{cases}$$

k=2:



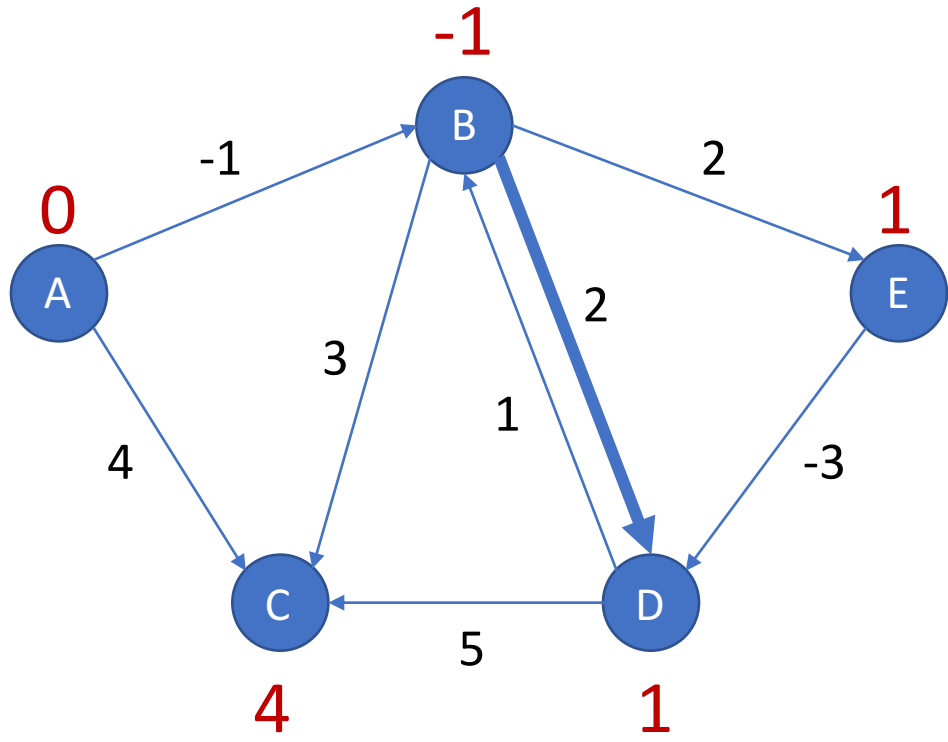
A	B	C	D	E
0	∞	∞	∞	∞
0	∞	∞	∞	∞
0	-1	∞	∞	∞
0	-1	4	∞	∞
0	-1	4	∞	∞
0	-1	4	∞	1

```
for each u in V
  d(u,0) = infty
d(s,0) = 0

for k = 1 to n-1
  for each v in V
    d(v,k) = d(v, k-1)
    for each edge in incoming(V)
      d(v,k) = min{d(v,k),
                  d(u,k-1)+l(u,v)}

for each v in V do
  dist(s,v) = d(v, n-1)
```

k=2:



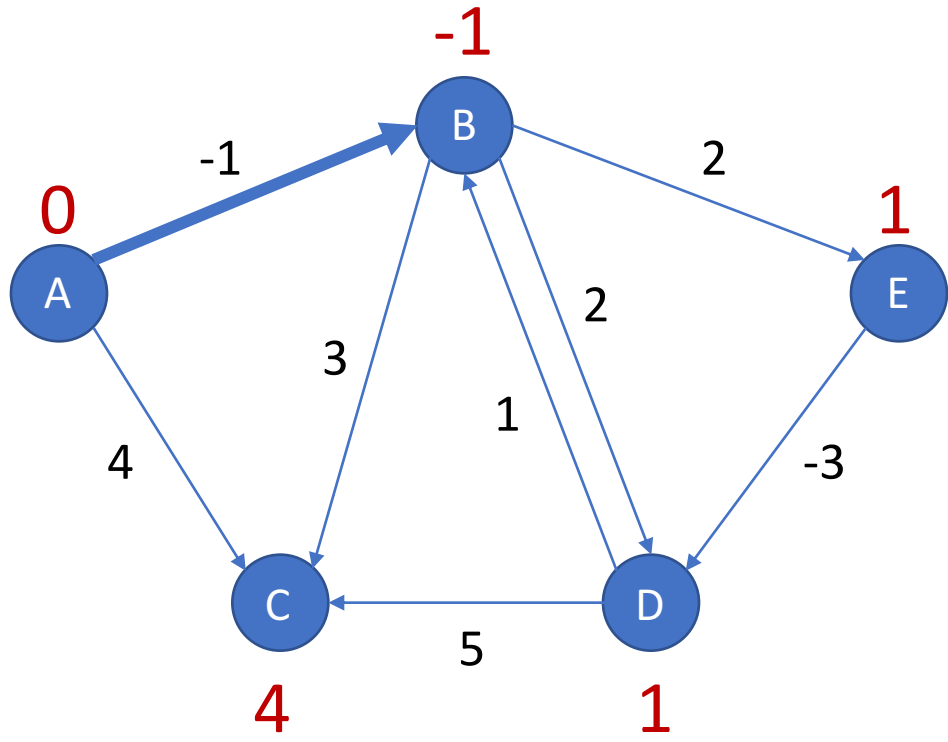
A	B	C	D	E
0	∞	∞	∞	∞
0	∞	∞	∞	∞
0	-1	∞	∞	∞
0	-1	4	∞	∞
0	-1	4	∞	∞
0	-1	4	∞	1
0	-1	4	1	1

```
for each u in V
  d(u,0) = inf
d(s,0) = 0

for k = 1 to n-1
  for each v in V
    d(v,k) = d(v, k-1)
    for each edge in incoming(V)
      d(v,k) = min{d(v,k),
                  d(u,k-1)+l(u,v)}

for each v in V do
  dist(s,v) = d(v, n-1)
```


k=2:



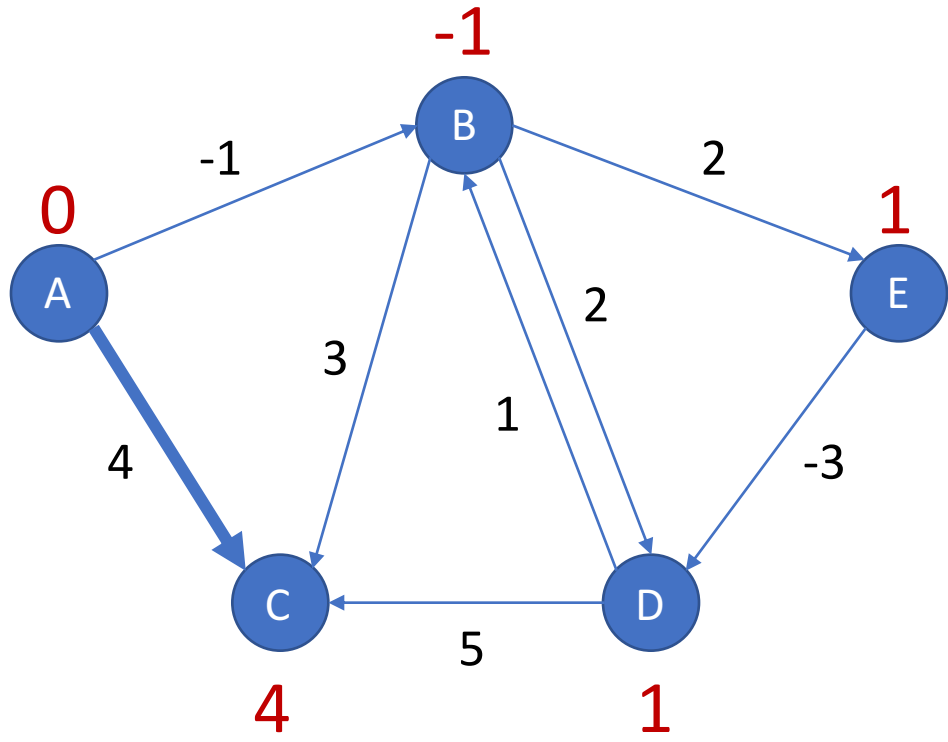
A	B	C	D	E
0	∞	∞	∞	∞
0	∞	∞	∞	∞
0	-1	∞	∞	∞
0	-1	4	∞	∞
0	-1	4	∞	∞
0	-1	4	∞	1
0	-1	4	1	1

```
for each u in V
  d(u,0) = inf
d(s,0) = 0

for k = 1 to n-1
  for each v in V
    d(v,k) = d(v, k-1)
    for each edge in incoming(V)
      d(v,k) = min{d(v,k),
                  d(u,k-1)+l(u,v)}

for each v in V do
  dist(s,v) = d(v, n-1)
```

k=2:



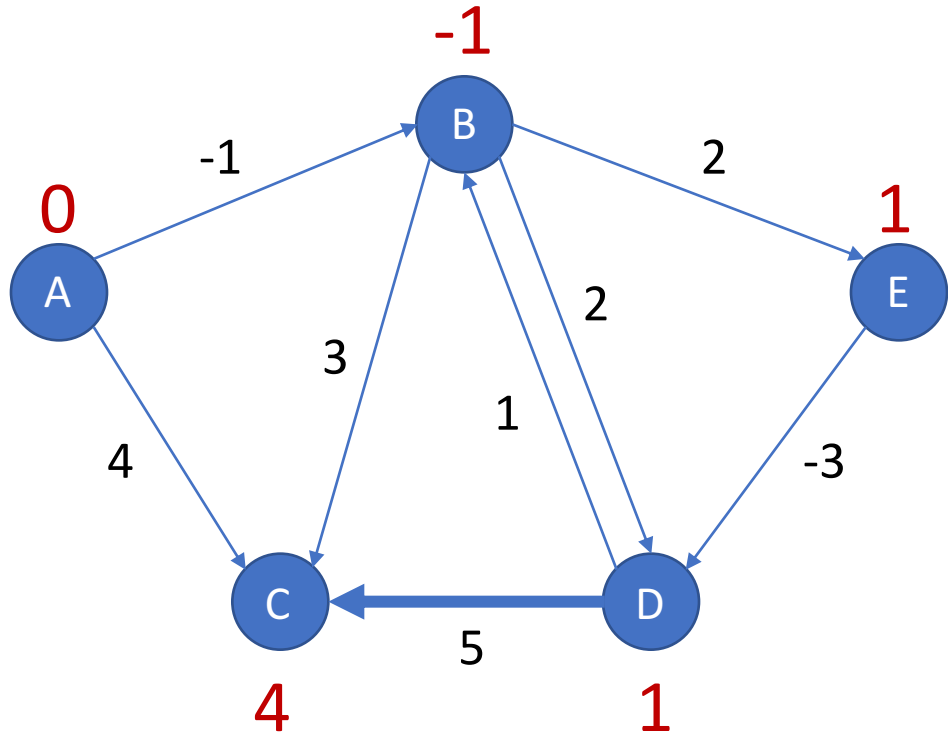
A	B	C	D	E
0	∞	∞	∞	∞
0	∞	∞	∞	∞
0	-1	∞	∞	∞
0	-1	4	∞	∞
0	-1	4	∞	∞
0	-1	4	∞	1
0	-1	4	1	1

```
for each u in V
  d(u,0) = inf
d(s,0) = 0

for k = 1 to n-1
  for each v in V
    d(v,k) = d(v, k-1)
    for each edge in incoming(V)
      d(v,k) = min{d(v,k),
                  d(u,k-1)+l(u,v)}

for each v in V do
  dist(s,v) = d(v, n-1)
```

k=2:



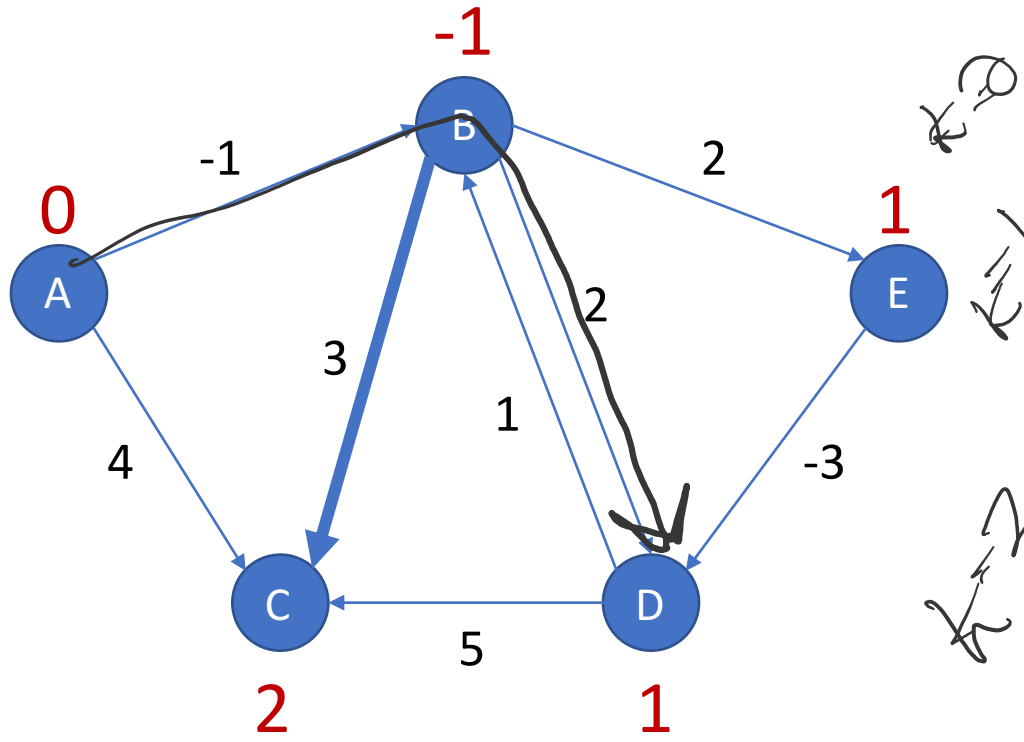
A	B	C	D	E
0	∞	∞	∞	∞
0	∞	∞	∞	∞
0	-1	∞	∞	∞
0	-1	4	∞	∞
0	-1	4	∞	∞
0	-1	4	∞	1
0	-1	4	1	1

```
for each u in V
  d(u,0) = inf
d(s,0) = 0

for k = 1 to n-1
  for each v in V
    d(v,k) = d(v, k-1)
    for each edge in incoming(V)
      d(v,k) = min{d(v,k),
                  d(u,k-1)+l(u,v)}

for each v in V do
  dist(s,v) = d(v, n-1)
```

k=2:



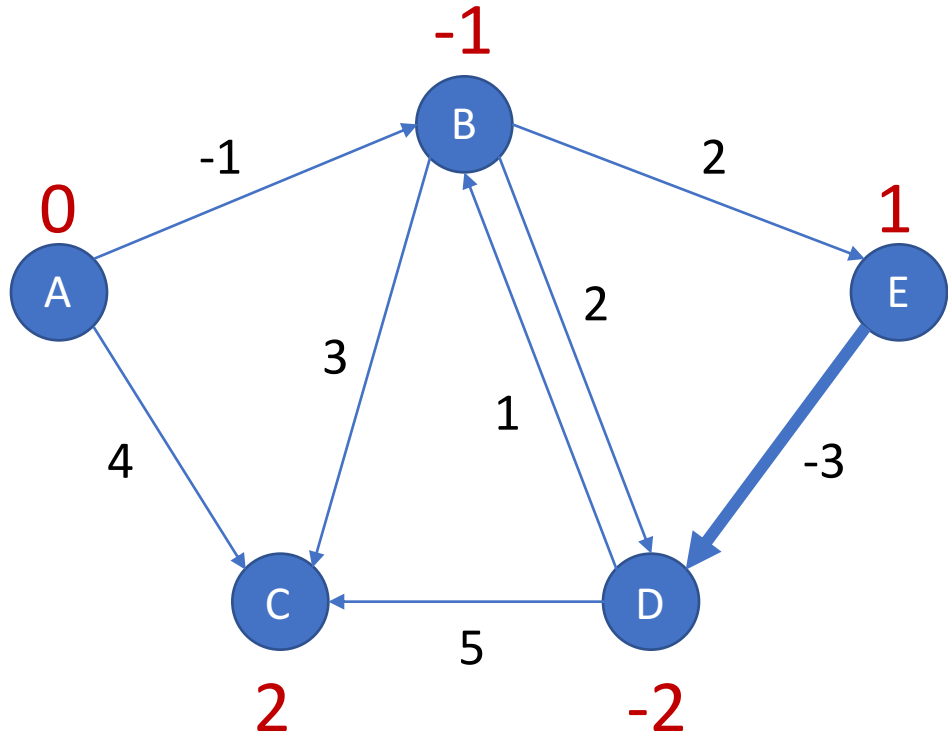
	A	B	C	D	E
k=0	0	∞	∞	∞	∞
k=1	0	∞	∞	∞	∞
k=2	0	-1	∞	∞	∞
k=3	0	-1	4	∞	∞
k=4	0	-1	4	∞	1
k=5	0	-1	4	1	1
k=6	0	-1	2	1	1

```
for each u in V
  d(u,0) = infity
d(s,0) = 0

for k = 1 to n-1
  for each v in V
    d(v,k) = d(v, k-1)
    for each edge in incoming(V)
      d(v,k) = min{d(v,k), d(u,k-1)+l(u,v)}

for each v in V do
  dist(s,v) = d(v, n-1)
```

k=2:



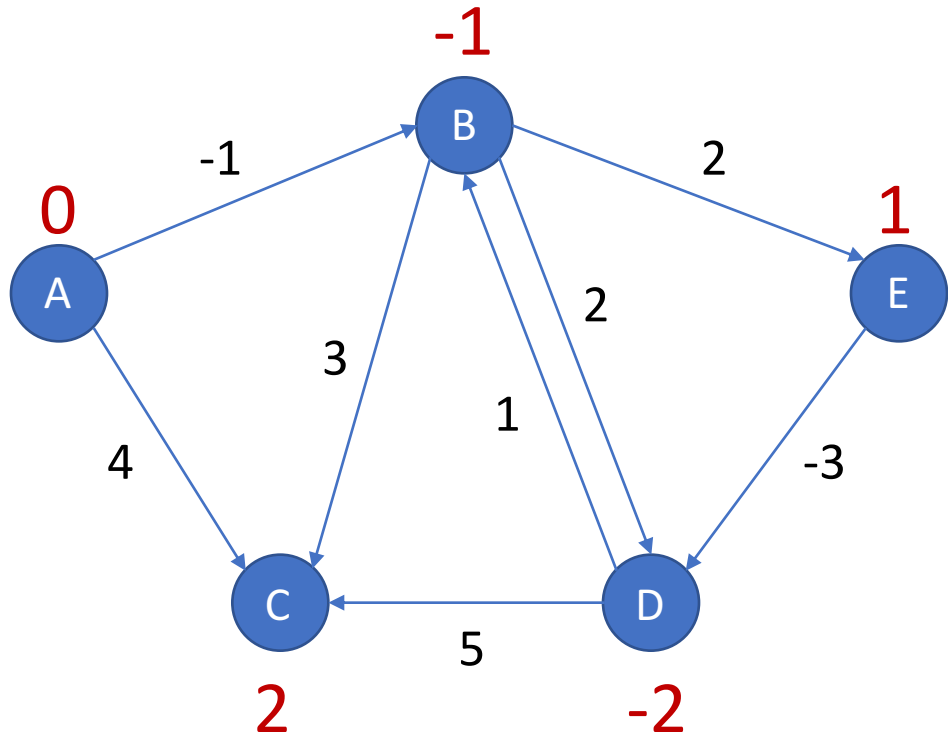
	A	B	C	D	E
A	0	∞	∞	∞	∞
B	0	∞	∞	∞	∞
C	0	-1	∞	∞	∞
D	0	-1	4	∞	∞
E	0	-1	4	∞	∞
	0	-1	4	∞	1
	0	-1	4	1	1
	0	-1	2	1	1
	0	-1	2	-2	1

```
for each u in V
  d(u,0) = inf
d(s,0) = 0

for k = 1 to n-1
  for each v in V
    d(v,k) = d(v, k-1)
    for each edge in incoming(V)
      d(v,k) = min{d(v,k),
                  d(u,k-1)+l(u,v)}

for each v in V do
  dist(s,v) = d(v, n-1)
```

k=2:



	A	B	C	D	E
A	0	∞	∞	∞	∞
B	0	∞	∞	∞	∞
C	0	-1	∞	∞	∞
D	0	-1	4	∞	∞
E	0	-1	4	∞	∞
	0	-1	4	∞	1
	0	-1	4	1	1
	0	-1	2	1	1
	0	-1	2	-2	1

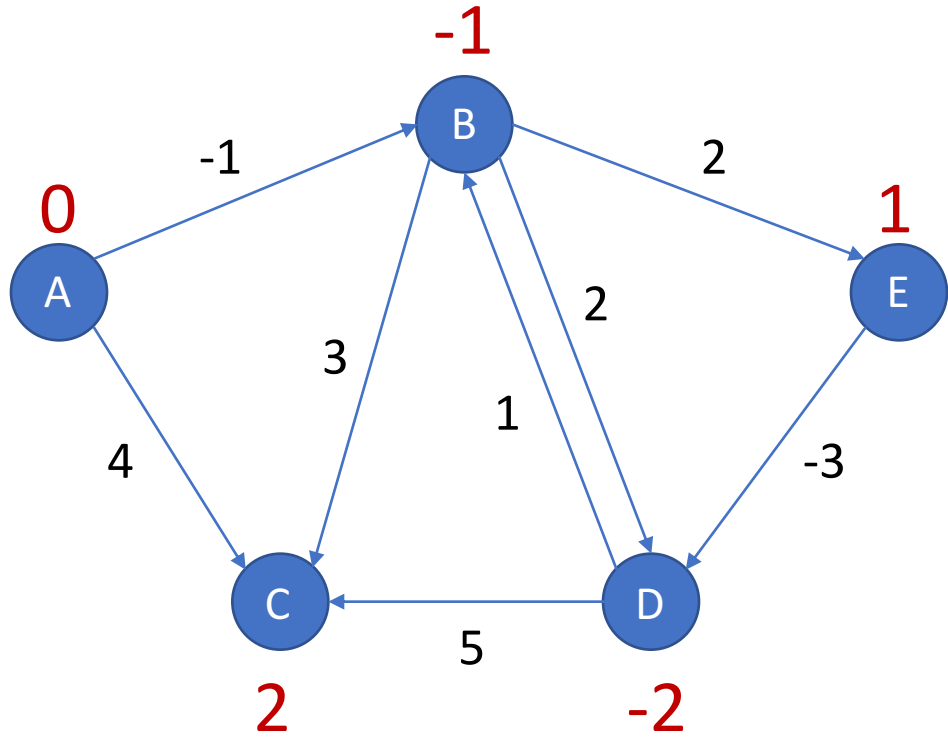
```
for each u in V
  d(u,0) = infty
d(s,0) = 0

for k = 1 to n-1
  for each v in V
    d(v,k) = d(v, k-1)
    for each edge in incoming(V)
      d(v,k) = min{d(v,k),
                  d(u,k-1)+l(u,v)}

for each v in V do
  dist(s,v) = d(v, n-1)
```

Distances assuming all paths lengths are at most **2 edges long**.

k=3:



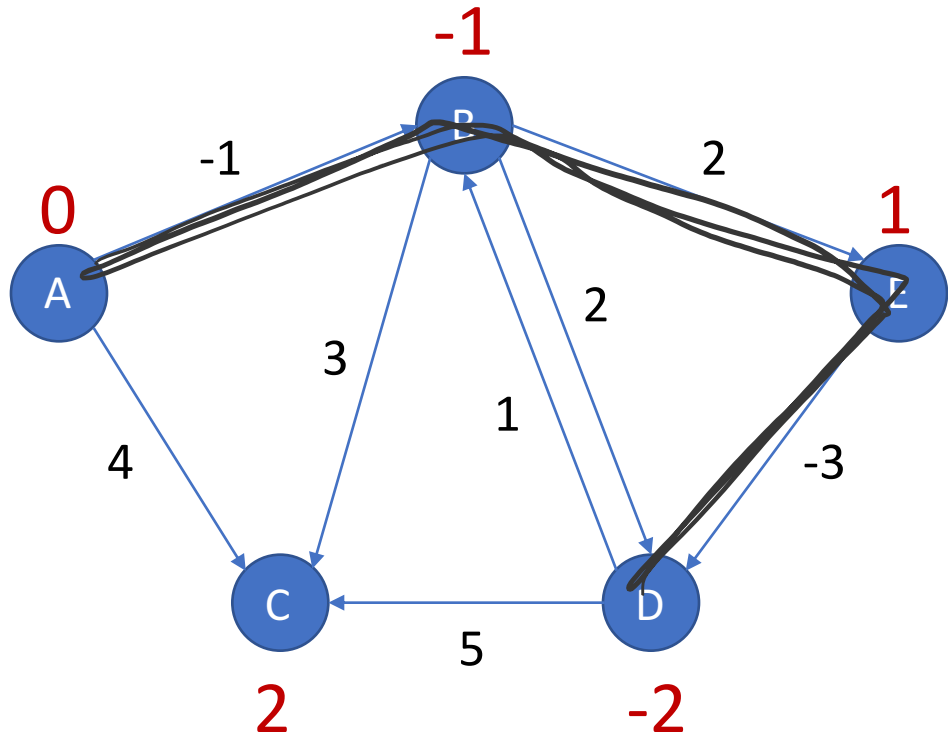
	A	B	C	D	E
A	0	∞	∞	∞	∞
B	0	∞	∞	∞	∞
C	0	-1	∞	∞	∞
D	0	-1	4	∞	∞
E	0	-1	4	∞	∞
	0	-1	4	1	1
	0	-1	2	1	1
	0	-1	2	-2	1
	0	-1	2	-2	1

```
for each u in V
  d(u,0) = infity
d(s,0) = 0

for k = 1 to n-1
  for each v in V
    d(v,k) = d(v, k-1)
    for each edge in incoming(V)
      d(v,k) = min{d(v,k),
                  d(u,k-1)+l(u,v)}

for each v in V do
  dist(s,v) = d(v, n-1)
```

k=4:



	A	B	C	D	E
A	0	∞	∞	∞	∞
B	0	∞	∞	∞	∞
C	0	-1	∞	∞	∞
D	0	-1	4	∞	∞
E	0	-1	4	∞	∞
A	0	-1	4	1	1
B	0	-1	2	1	1
C	0	-1	2	-2	1
D	0	-1	2	-2	1
E	0	-1	2	-2	1

for each u in V
 $d(u,0) = \text{infty}$
 $d(s,0) = 0$

for k = 1 to n-1
 for each v in V
 $d(v,k) = d(v, k-1)$
 for each edge in incoming(v)
 $d(v,k) = \min\{d(v,k), d(u,k-1) + l(u,v)\}$

for each v in V do
 $\text{dist}(s,v) = d(v, n-1)$

for k = n

$O(n^2)$

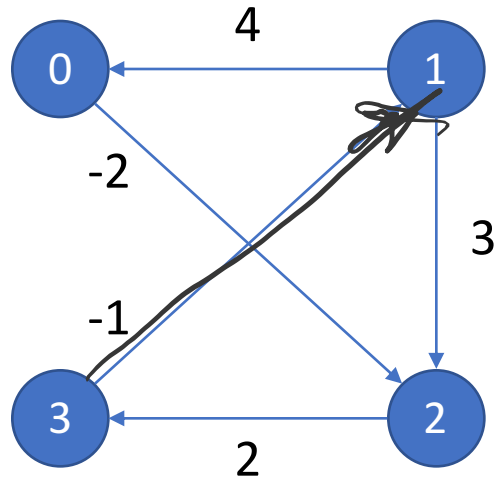
$n^2 = n^2$

$O(n^3)$

Floyd-Warshall Algorithm Example

Initialization:

$$A^1 = \begin{bmatrix} 0 & \infty & -2 & \infty \\ 4 & 0 & 3 & \infty \\ \infty & \infty & 0 & 2 \\ \infty & -1 & \infty & 0 \end{bmatrix}$$



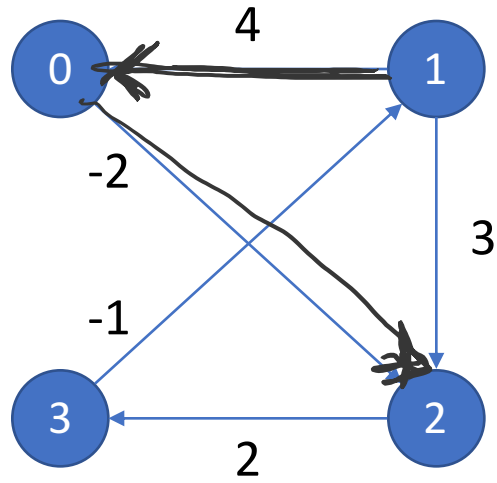
```
for i = 1 to n
  for j = 1 to n
    d(i,j,0) = l(i,j)
```

Base Case

```
for k = 0 to n ←
  for i = 0 to n
    for j = 0 to n
      d(i,j,k) = min{d(i,j,k-1),
                    d(i,k,k-1)+d(k,j,k-1)}
```

```
for i = 0 to n
  if dist(i,i,n) < 0
    print("Negative cycle detected!")
```

k=0:



$A^1 =$

$$\begin{bmatrix} 0 & \infty & -2 & \infty \\ 4 & 0 & 3 & \infty \\ \infty & \infty & 0 & 2 \\ \infty & -1 & \infty & 0 \end{bmatrix}$$

$A^0 =$

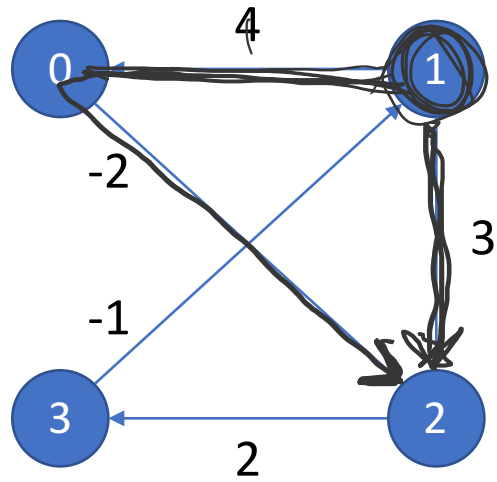
$$\begin{bmatrix} 0 & \infty & -2 & \infty \\ 4 & 0 & 2 & \infty \\ \infty & \infty & 0 & 2 \\ \infty & -1 & \infty & 0 \end{bmatrix}$$

```
for i = 1 to n
  for j = 1 to n
    d(i,j,0) = l(i,j)
```

```
for k = 0 to n
  for i = 0 to n
    for j = 0 to n
      d(i,j,k) = min{d(i,j,k-1),
                    d(i,k,k-1)+d(k,j,k-1)}
```

```
for i = 0 to n
  if dist(i,i,n) < 0
    print("Negative cycle detected!")
```

k=0:



$$A^1 = \begin{bmatrix} 0 & \infty & -2 & \infty \\ 4 & 0 & 3 & \infty \\ \infty & \infty & 0 & 2 \\ \infty & -1 & \infty & 0 \end{bmatrix}$$

$$A^0 = \begin{bmatrix} 0 & \infty & -2 & \infty \\ 4 & 0 & 2 & \infty \\ \infty & \infty & 0 & 2 \\ \infty & -1 & \infty & 0 \end{bmatrix}$$

$$d(2,3,0) = \min \begin{cases} d(2,3,1) = 2 \\ d(2,0,1) + d(0,3,1) = \infty + \infty = \infty \end{cases}$$

$$d(1,2,0) = \min \begin{cases} d(1,2,1) = 3 \\ d(1,0,1) + d(0,2,1) = 4 + (-2) = 2 \end{cases}$$

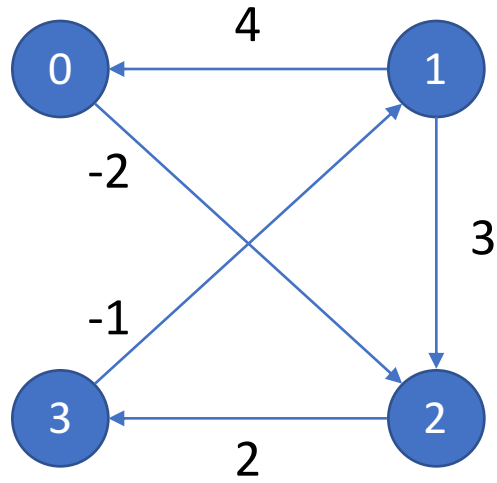
```

for i = 1 to n
  for j = 1 to n
    d(i,j,0) = l(i,j)

for k = 0 to n
  for i = 0 to n
    for j = 0 to n
      d(i,j,k) = min{d(i,j,k-1),
                    d(i,k,k-1)+d(k,j,k-1)}

for i = 0 to n
  if dist(i,i,n) < 0
    print("Negative cycle detected!")
  
```

k=1:



$$A^1 = \begin{bmatrix} 0 & \infty & -2 & \infty \\ 4 & 0 & 3 & \infty \\ \infty & \infty & 0 & 2 \\ \infty & -1 & \infty & 0 \end{bmatrix}$$

$$A^0 = \begin{bmatrix} 0 & \infty & -2 & \infty \\ 4 & 0 & 2 & \infty \\ \infty & \infty & 0 & 2 \\ \infty & -1 & \infty & 0 \end{bmatrix}$$

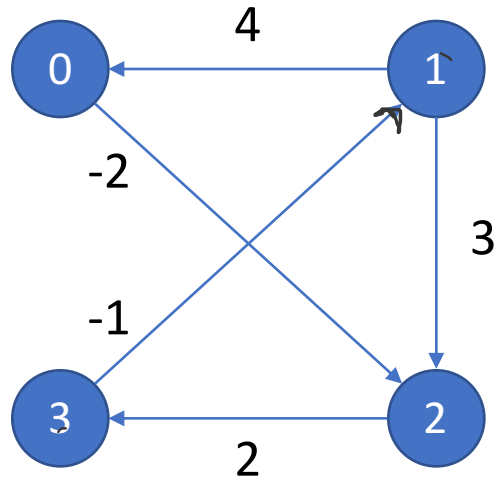
$$A^1 = \begin{bmatrix} 0 & \infty & -2 & \infty \\ 4 & 0 & 2 & \infty \\ \infty & \infty & 0 & 2 \\ 3 & -1 & 1 & 0 \end{bmatrix}$$

```
for i = 1 to n
  for j = 1 to n
    d(i,j,0) = l(i,j)
```

```
for k = 0 to n
  for i = 0 to n
    for j = 0 to n
      d(i,j,k) = min{d(i,j,k-1),
                    d(i,k,k-1)+d(k,j,k-1)}
```

```
for i = 0 to n
  if dist(i,i,n) < 0
    print("Negative cycle detected!")
```

k=1:



$$A^1 = \begin{bmatrix} 0 & \infty & -2 & \infty \\ 4 & 0 & 3 & \infty \\ \infty & \infty & 0 & 2 \\ \infty & -1 & \infty & 0 \end{bmatrix}$$

$$A^0 = \begin{bmatrix} 0 & \infty & -2 & \infty \\ 4 & 0 & 2 & \infty \\ \infty & \infty & 0 & 2 \\ \infty & -1 & \infty & 0 \end{bmatrix}$$

$$A^1 = \begin{bmatrix} 0 & \infty & -2 & \infty \\ 4 & 0 & 2 & \infty \\ \infty & \infty & 0 & 2 \\ \mathbf{3} & -1 & \mathbf{1} & 0 \end{bmatrix}$$

```
for i = 1 to n
  for j = 1 to n
    d(i,j,0) = l(i,j)

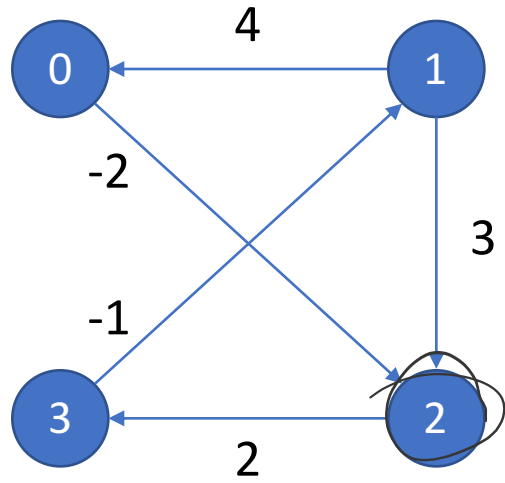
for k = 0 to n
  for i = 0 to n
    for j = 0 to n
      d(v,k) = min{d(i,j,k-1),
                  d(i,k,k-1)+d(k,j,k-1)}

for i = 0 to n
  if dist(i,i,n) < 0
    print("Negative cycle detected!")
```

$d(3,0,1) = \min \left\{ \begin{array}{l} d(3,0,0) = \infty \\ d(3,1,0) + d(1,0,0) = (-1) + (4) = 3 \end{array} \right.$

(Note: In the original image, the value 3 in the second term is circled, and an arrow points from the circled 3 in the matrix above to this calculation.)

k=2:



$$A^1 = \begin{bmatrix} 0 & \infty & -2 & \infty \\ 4 & 0 & 3 & \infty \\ \infty & \infty & 0 & 2 \\ \infty & -1 & \infty & 0 \end{bmatrix}$$

$$A^0 = \begin{bmatrix} 0 & \infty & -2 & \infty \\ 4 & 0 & 2 & \infty \\ \infty & \infty & 0 & 2 \\ \infty & -1 & \infty & 0 \end{bmatrix}$$

$$A^1 = \begin{bmatrix} 0 & \infty & -2 & \infty \\ 4 & 0 & 2 & \infty \\ \infty & \infty & 0 & 2 \\ 3 & -1 & 1 & 0 \end{bmatrix}$$

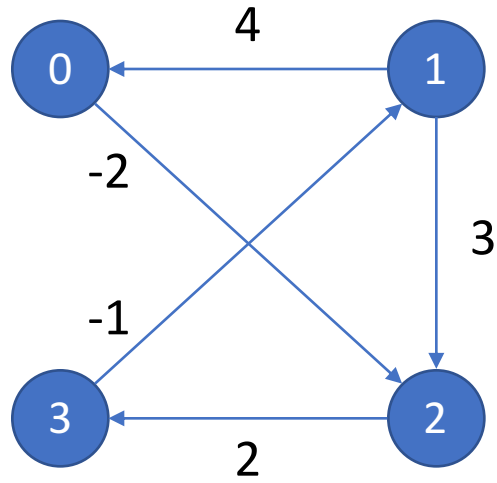
```
for i = 1 to n
  for j = 1 to n
    d(i,j,0) = l(i,j)
```

```
for k = 0 to n
  for i = 0 to n
    for j = 0 to n
      d(v,k) = min{d(i,j,k-1),
                  d(i,k,k-1)+d(k,j,k-1)}
```

```
for i = 0 to n
  if dist(i,i,n) < 0
    print("Negative cycle detected!")
```

$$A^2 = \begin{bmatrix} 0 & \infty & -2 & 0 \\ 4 & 0 & 2 & 4 \\ \infty & \infty & 0 & 2 \\ 3 & -1 & 1 & 0 \end{bmatrix}$$

k=3:



$$A^1 = \begin{bmatrix} 0 & \infty & -2 & \infty \\ 4 & 0 & 3 & \infty \\ \infty & \infty & 0 & 2 \\ \infty & -1 & \infty & 0 \end{bmatrix}$$

$$A^0 = \begin{bmatrix} 0 & \infty & -2 & \infty \\ 4 & 0 & 2 & \infty \\ \infty & \infty & 0 & 2 \\ \infty & -1 & \infty & 0 \end{bmatrix}$$

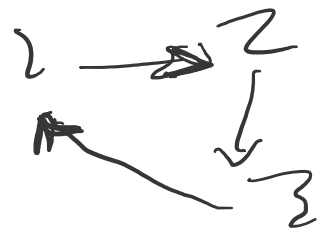
$$A^1 = \begin{bmatrix} 0 & \infty & -2 & \infty \\ 4 & 0 & 2 & \infty \\ \infty & \infty & 0 & 2 \\ 3 & -1 & 1 & 0 \end{bmatrix}$$

```

for i = 1 to n
  for j = 1 to n
    d(i,j,0) = l(i,j)

for k = 0 to n
  for i = 0 to n
    for j = 0 to n
      d(i,j,k) = min{d(i,j,k-1),
                    d(i,k,k-1)+d(k,j,k-1)}

for i = 0 to n
  if dist(i,i,n) < 0
    print("Negative cycle detected!")
  
```



$$A^2 = \begin{bmatrix} 0 & \infty & -2 & 0 \\ 4 & 0 & 2 & 4 \\ \infty & \infty & 0 & 2 \\ 3 & -1 & 1 & 0 \end{bmatrix}$$

$$A^3 = \begin{bmatrix} 0 & -1 & -2 & 0 \\ 4 & 0 & 2 & 4 \\ 5 & 1 & 0 & 2 \\ 3 & -1 & 1 & 0 \end{bmatrix}$$