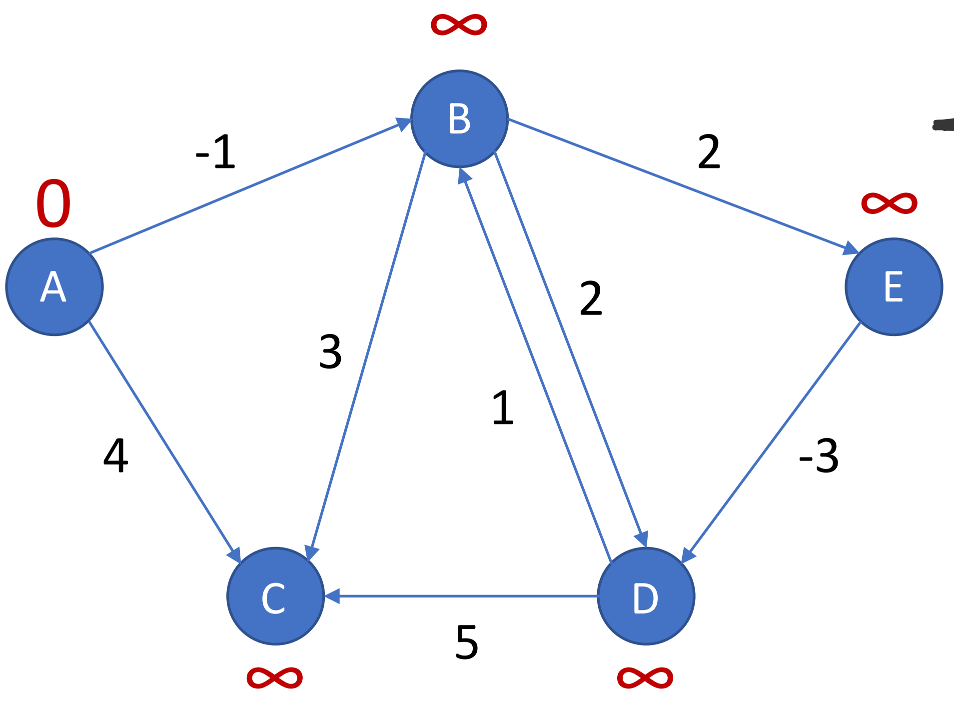
Bellman-Ford Algorithm Example

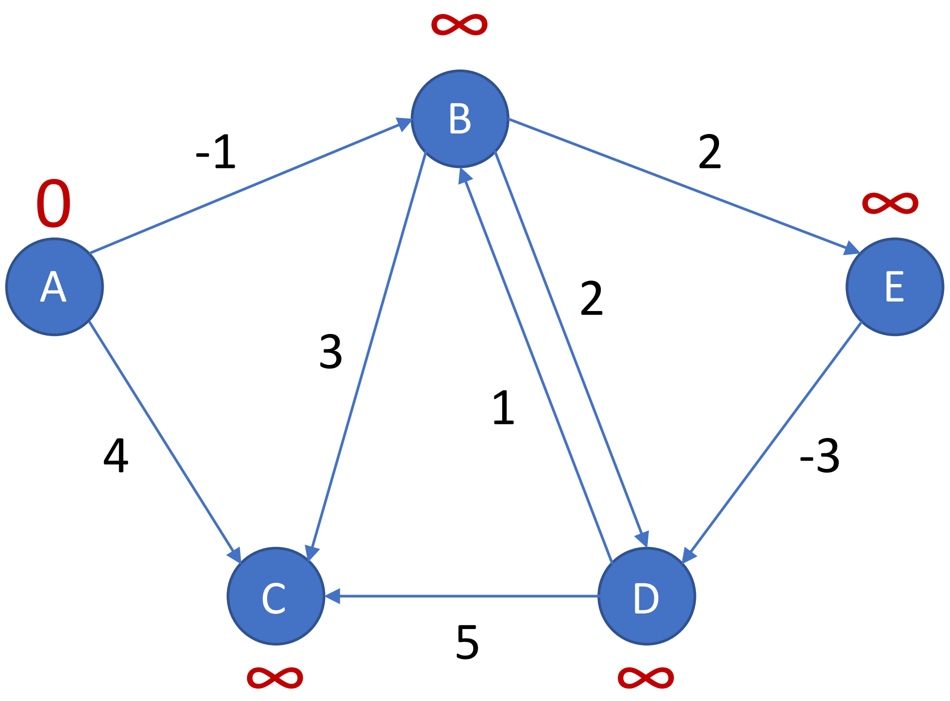
Initialization:



A	B	C	D	E
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)
```

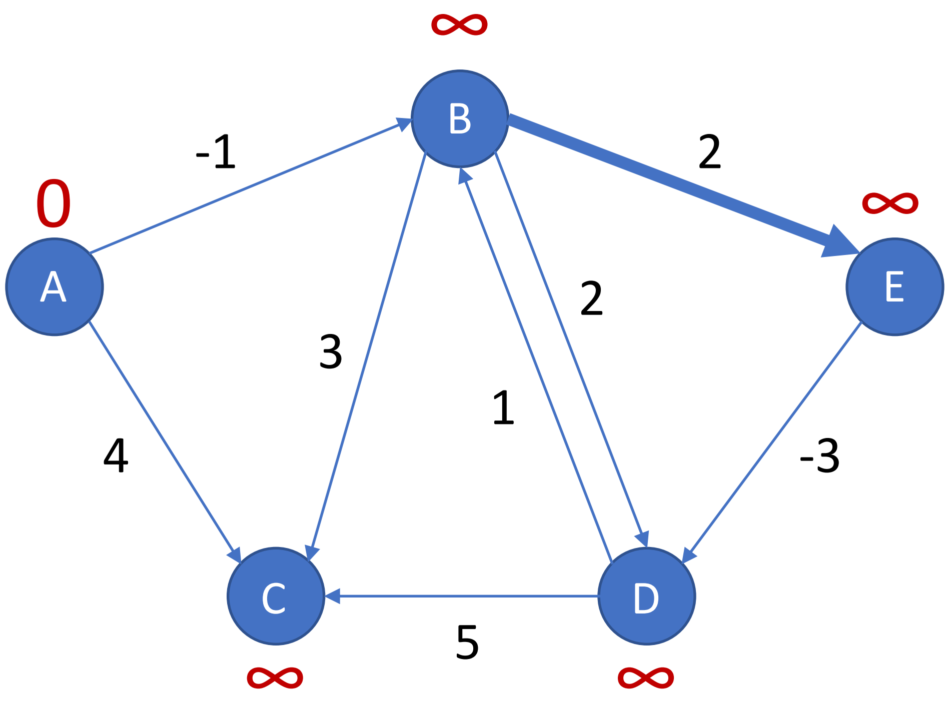
k=1:



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:



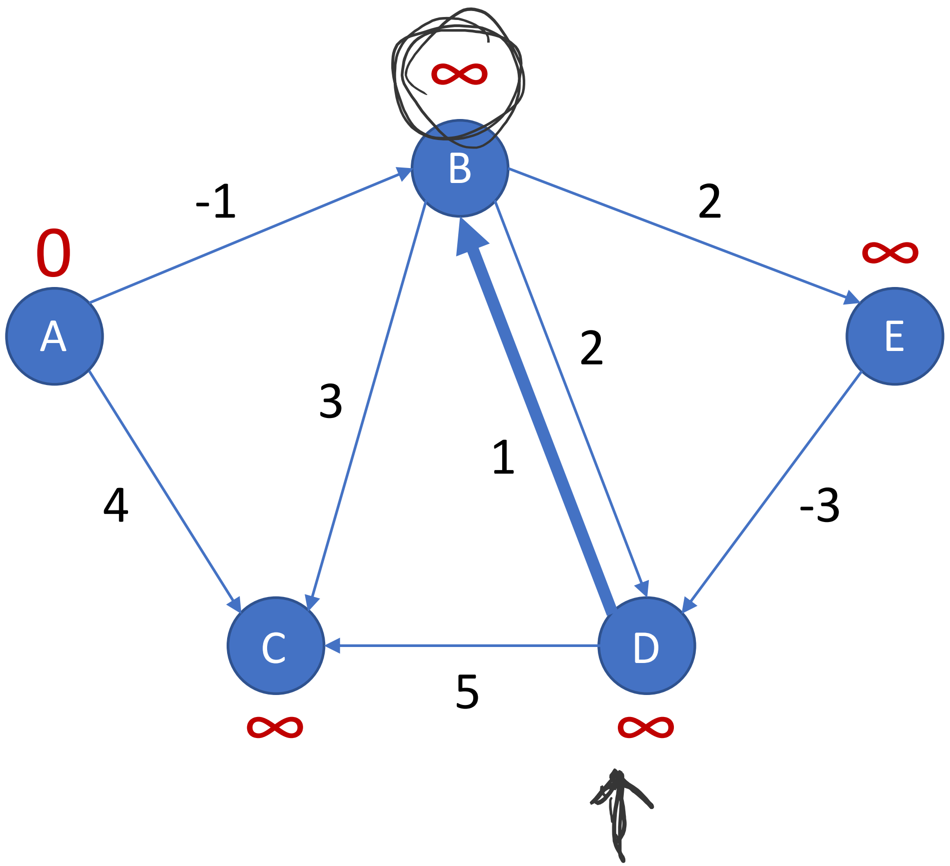
$d(v, \ )$

	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)
```

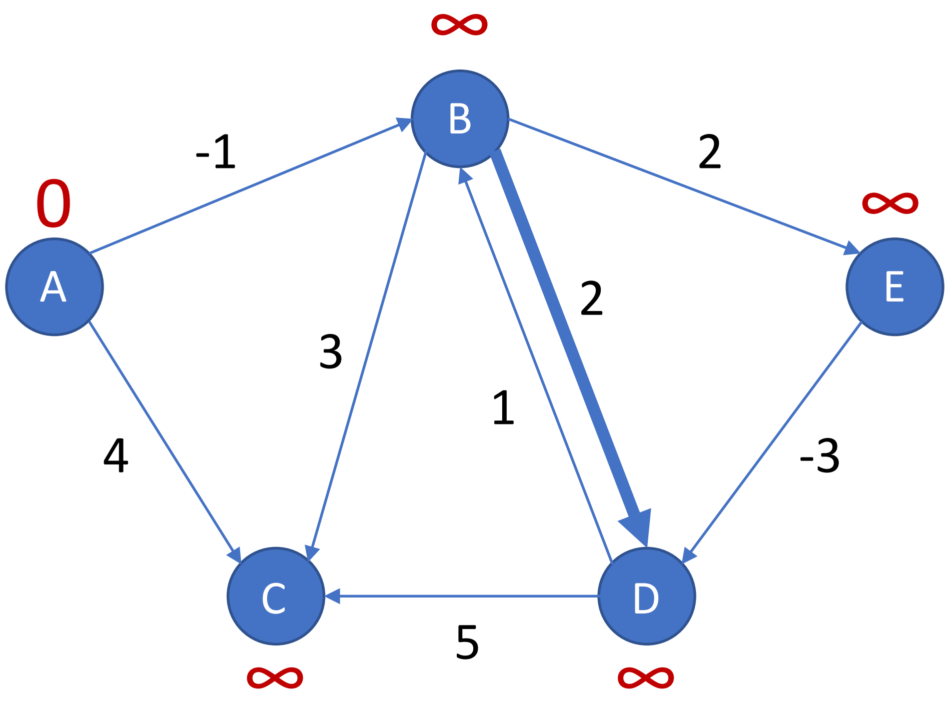
k=1:



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)
```

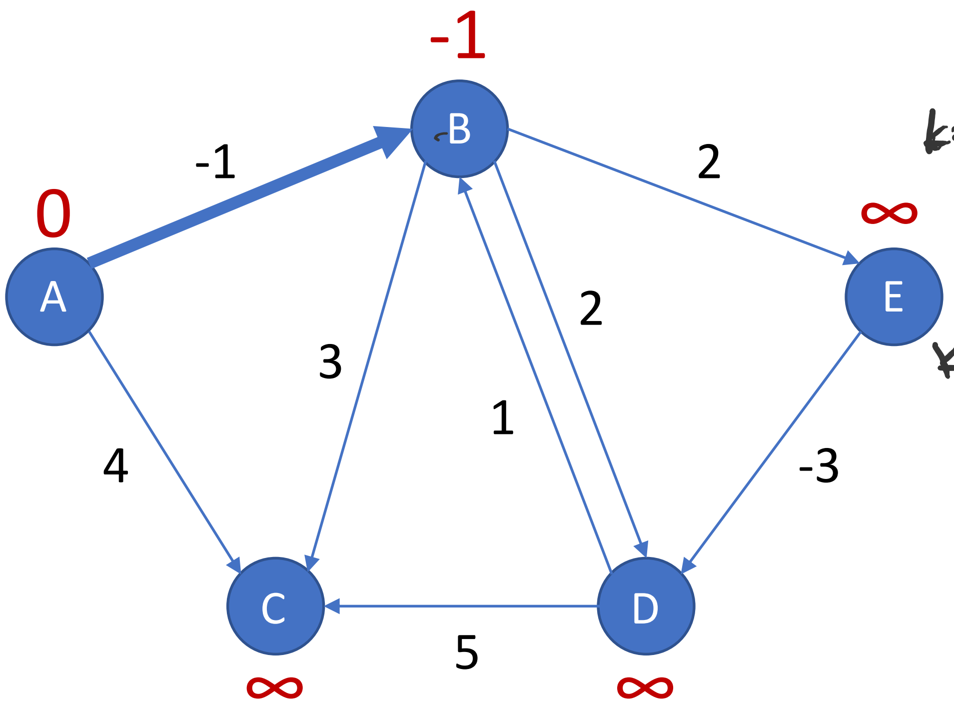
k=1:



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:

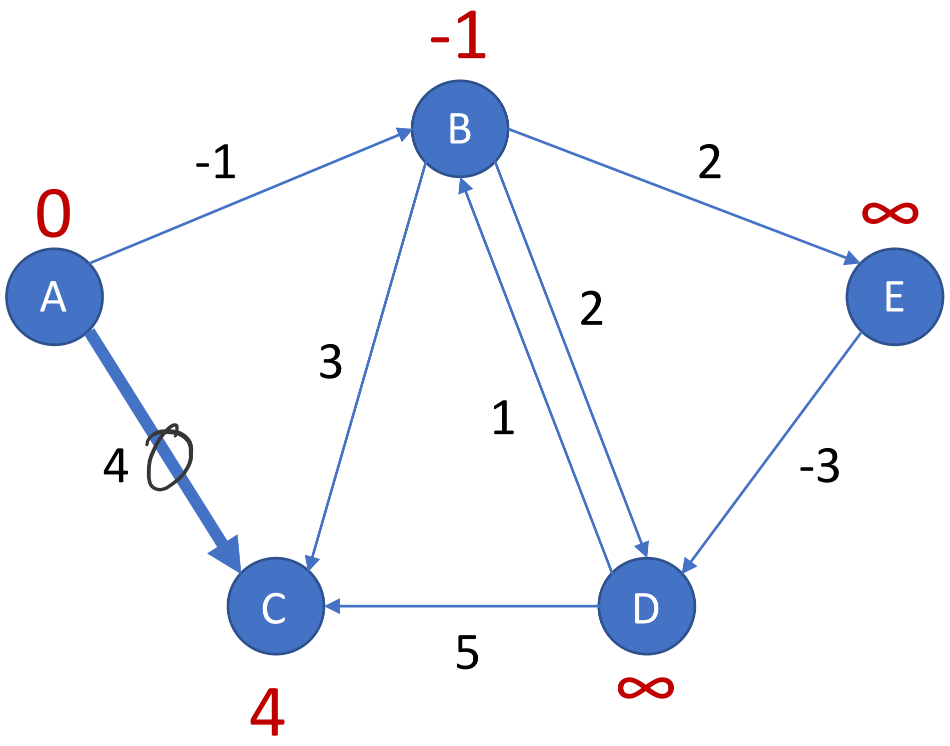


	A	B	C	D	E
k=0	0	∞	∞	∞	∞
	0	∞	∞	∞	∞
k=1	0	-1	∞	∞	∞

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

```
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)
```

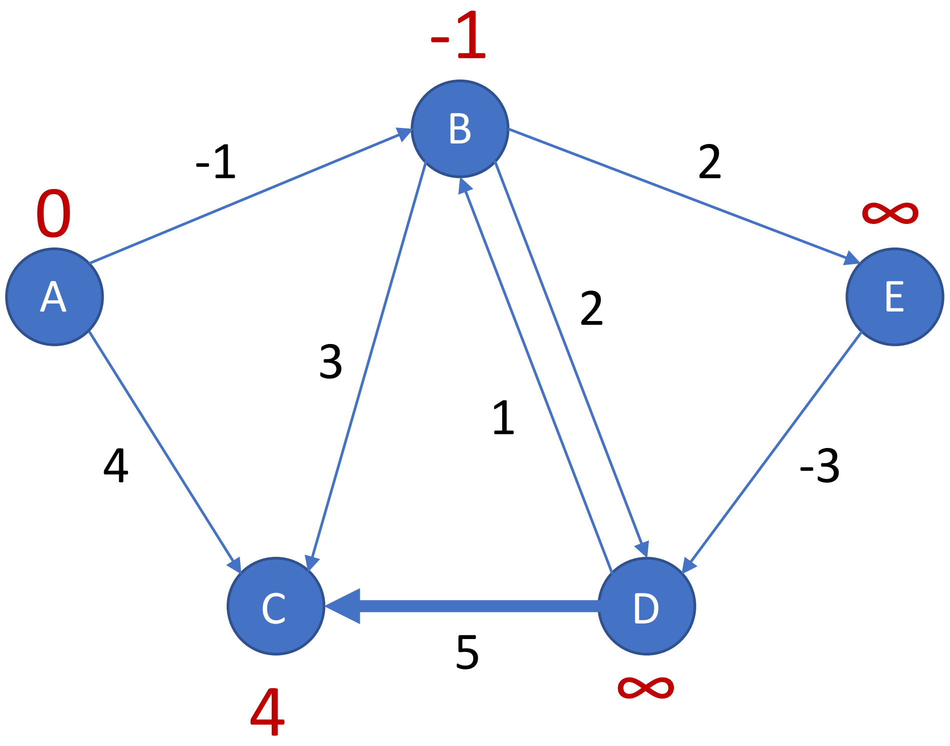
k=1:



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)
```

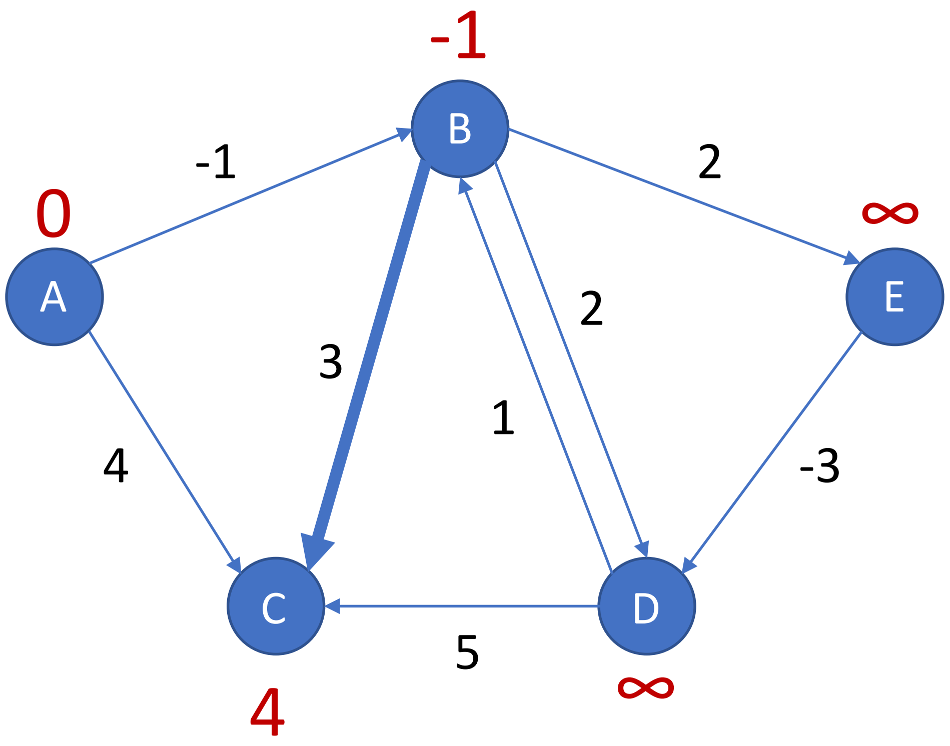
k=1:



A	B	C	D	E
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)
```

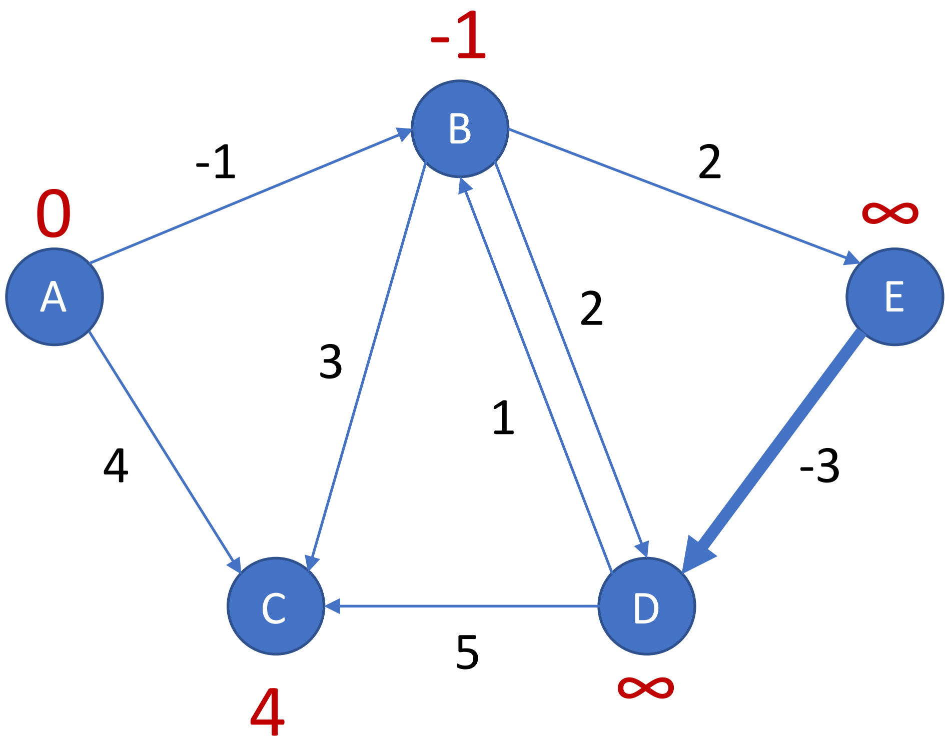
k=1:



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)
```

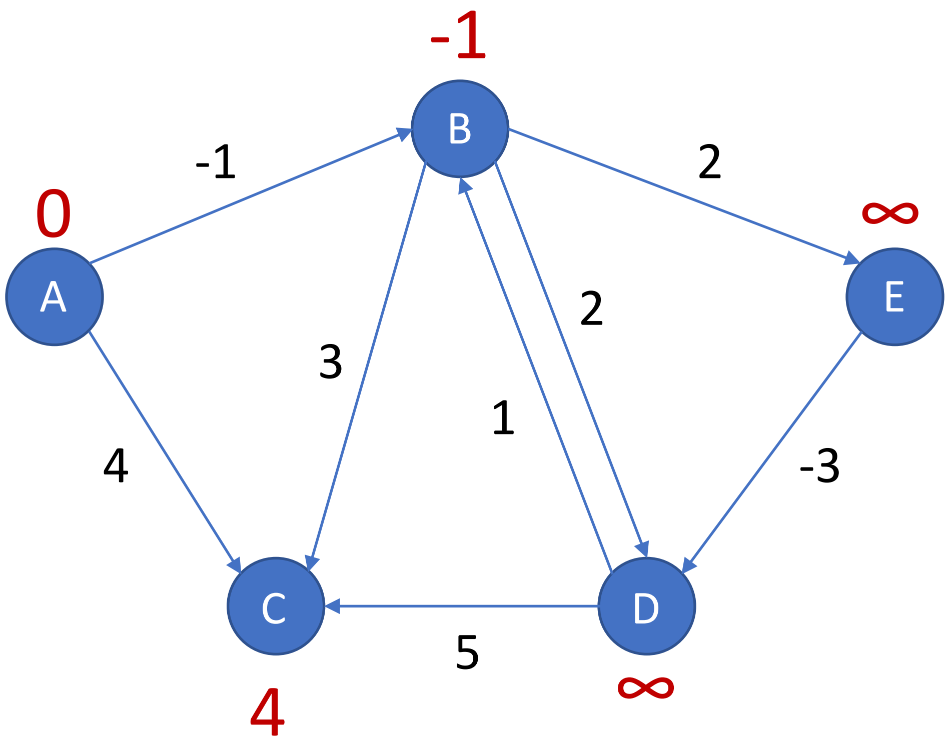
k=1:



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:

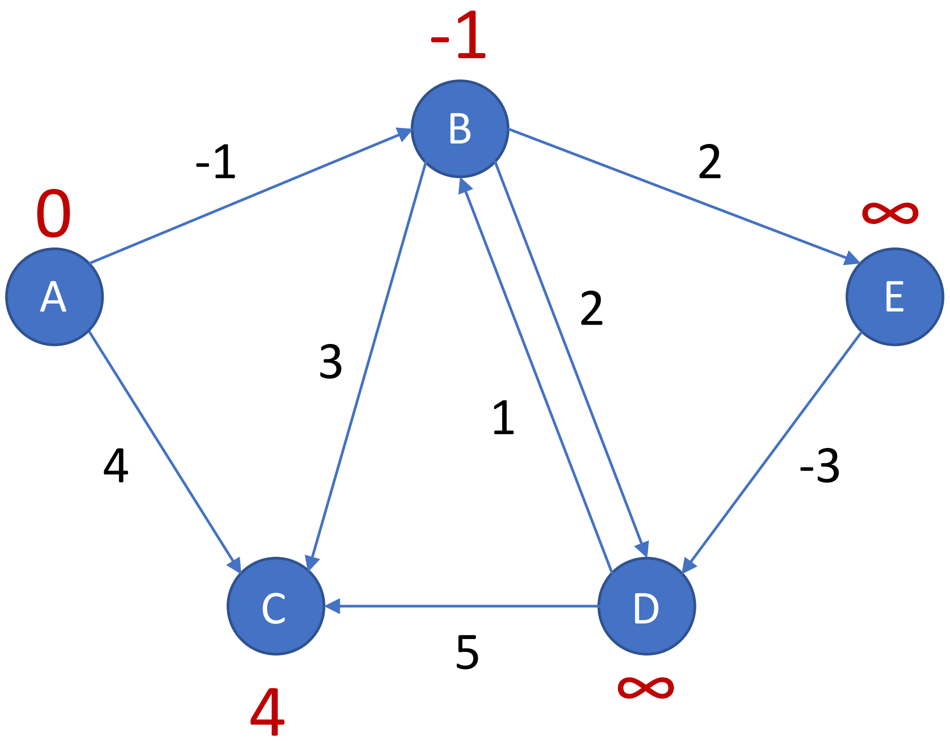


A	B	C	D	E
0	∞	∞	∞	∞
0	∞	∞	∞	∞
0	-1	∞	∞	∞
0	-1	4	∞	∞

```
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*.

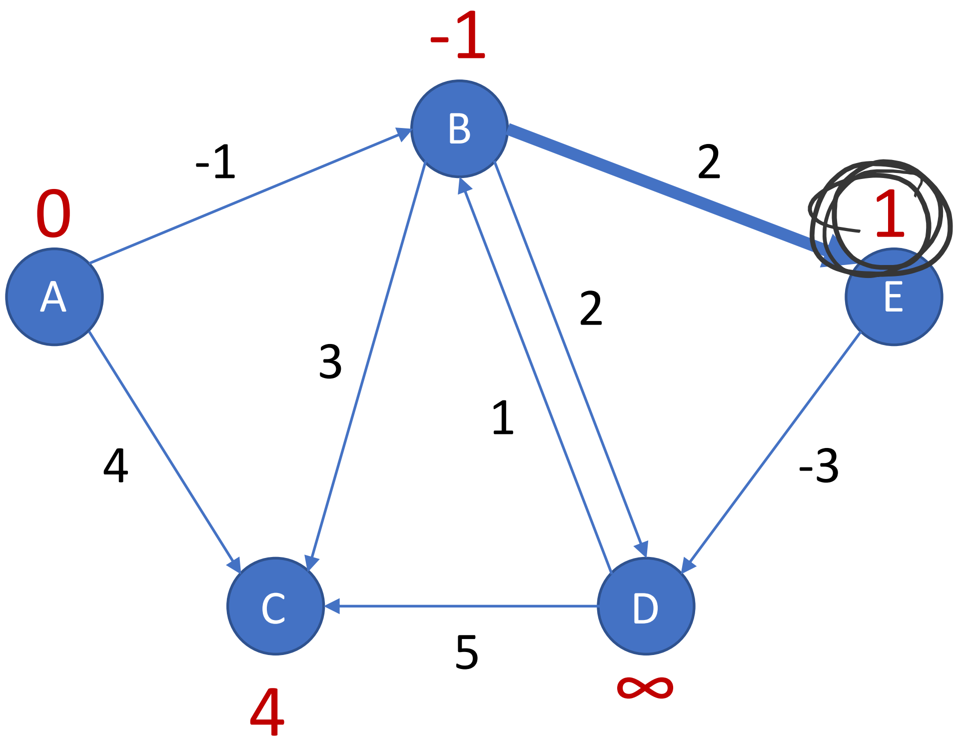
k=2:



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



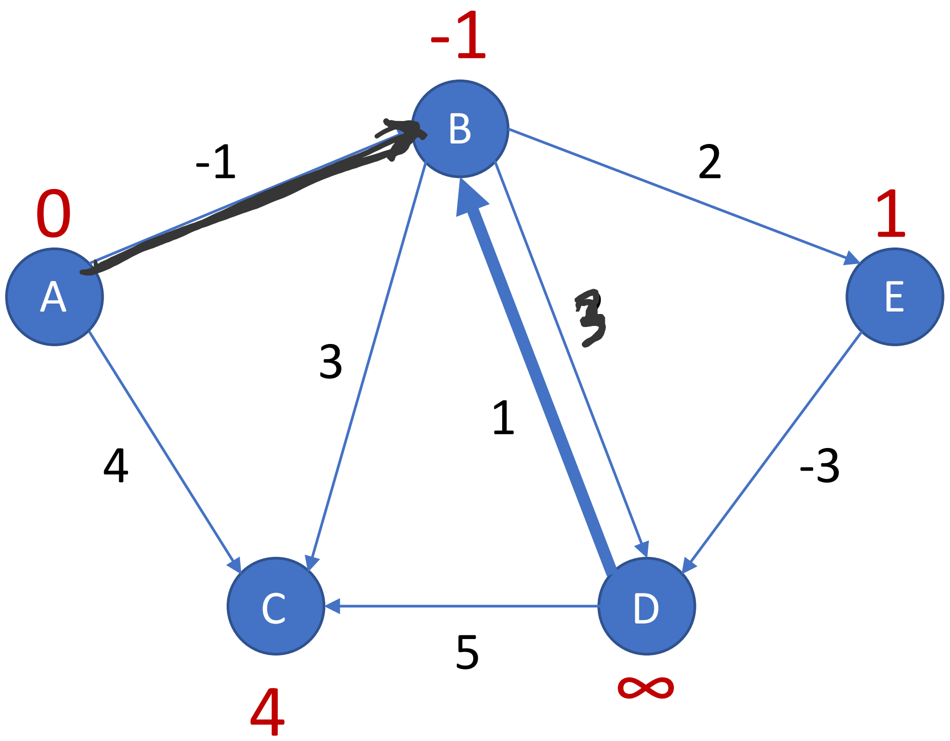
A	B	C	D	E
0	∞	∞	∞	∞
0	∞	∞	∞	∞
0	-1	∞	∞	∞
0	-1	4	∞	∞
0	-1	4	∞	∞
0	-1	4	∞	1

```
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)
```

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

-1

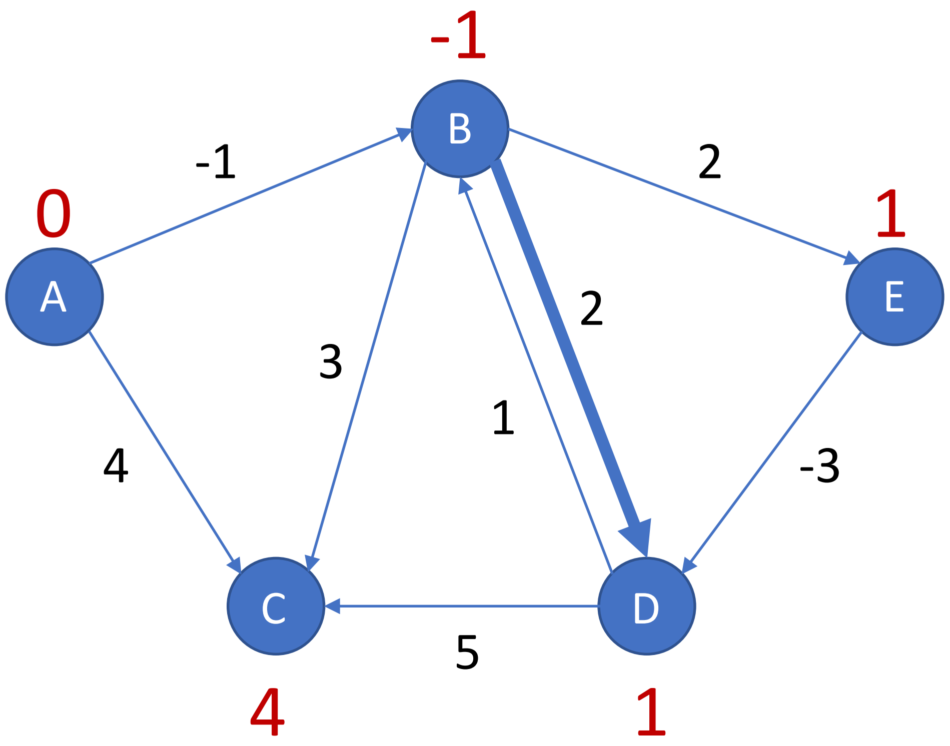
k=2:



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) = 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)
```

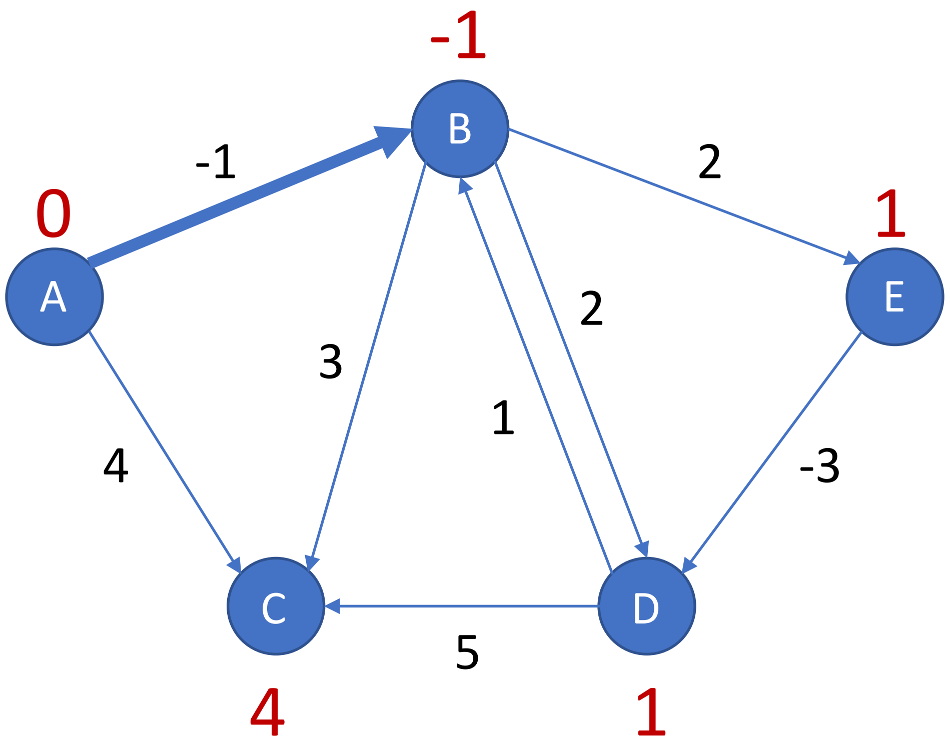
k=2:



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) = 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)
```

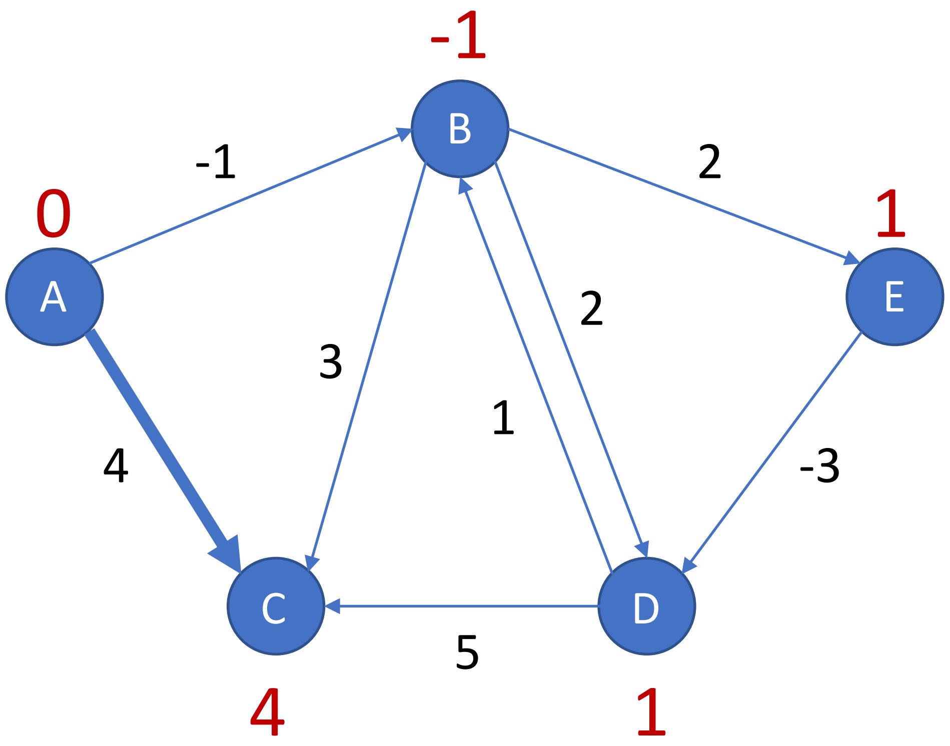
k=2:



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) = 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)
```

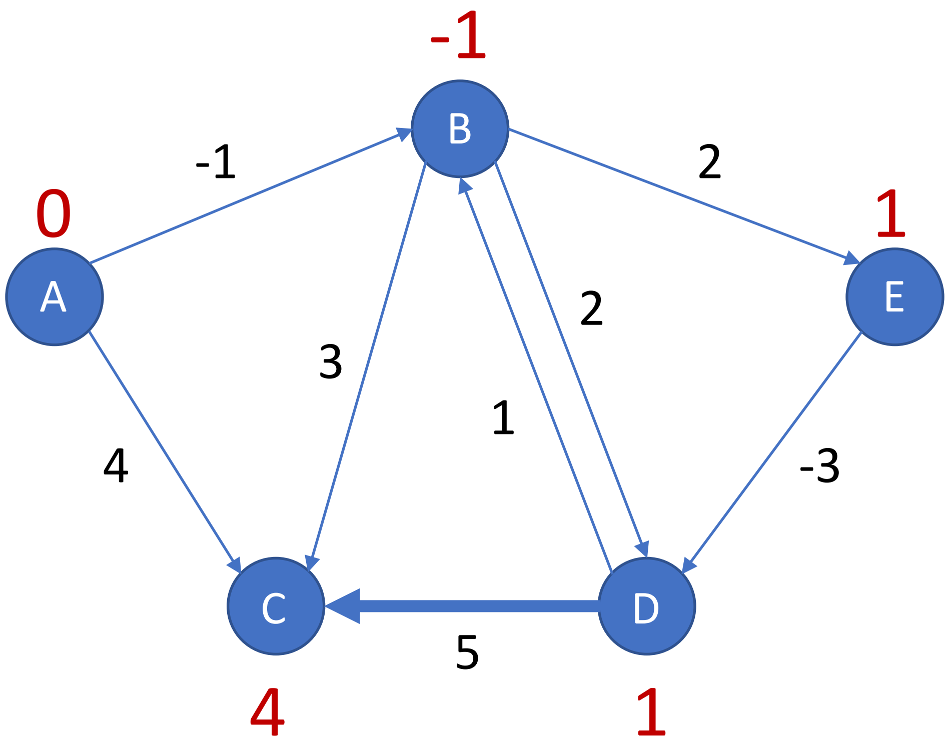
k=2:



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) = 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)
```

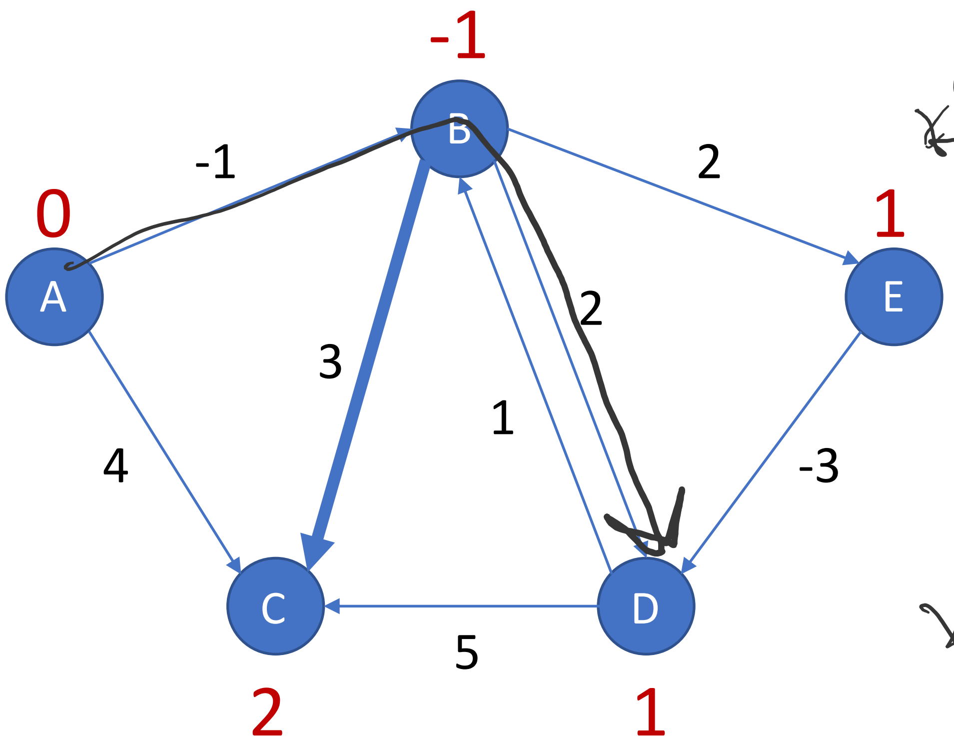
k=2:



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) = 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)
```

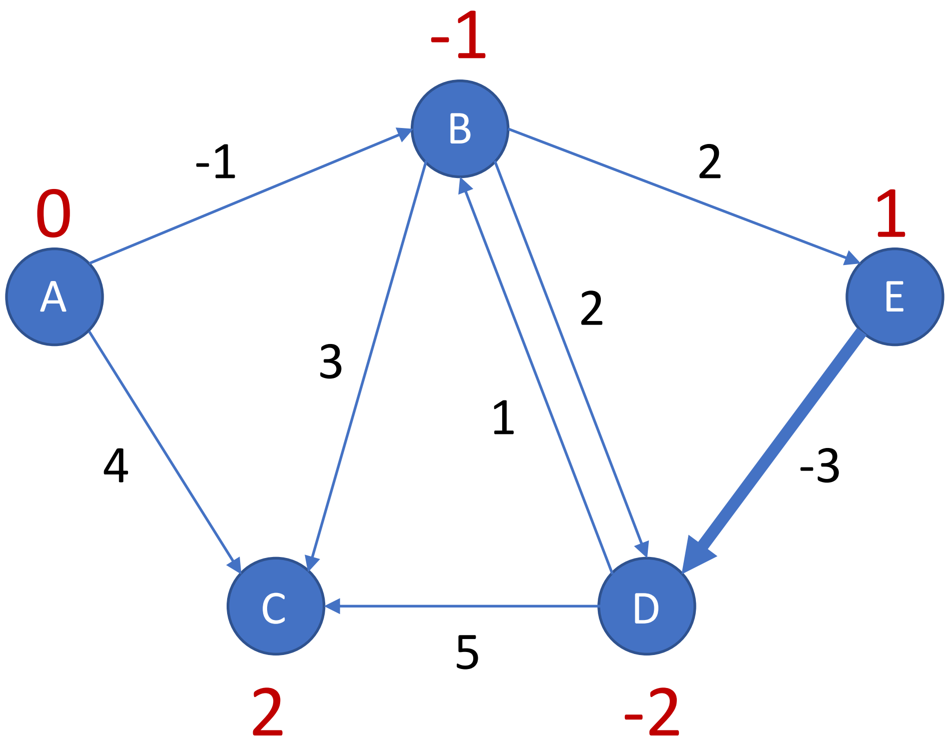
k=2:



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

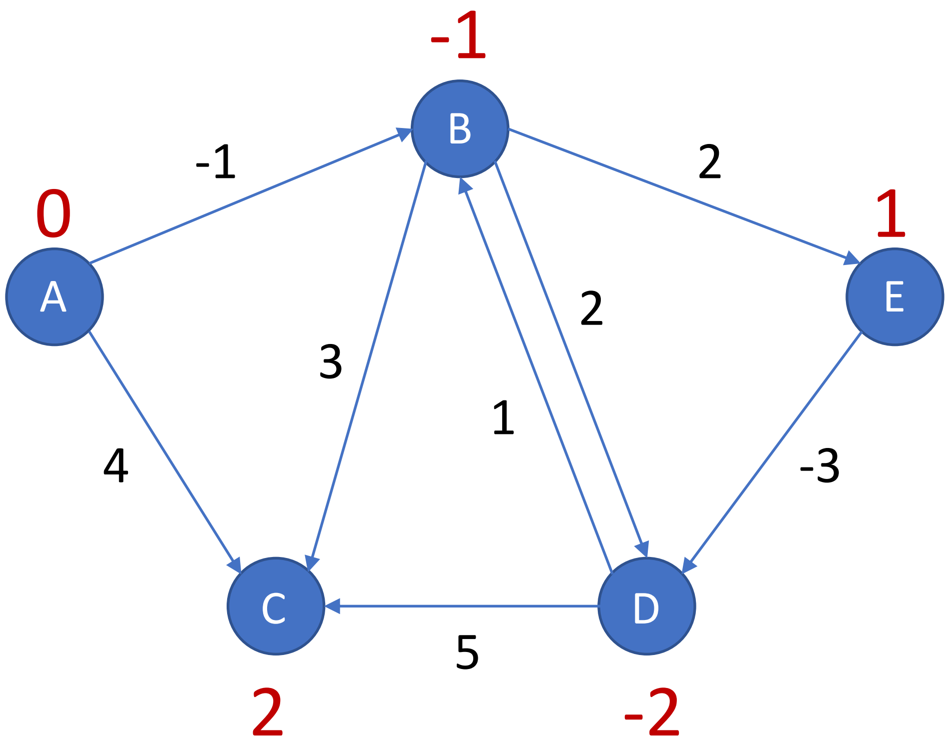
k=2:



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

k=2:

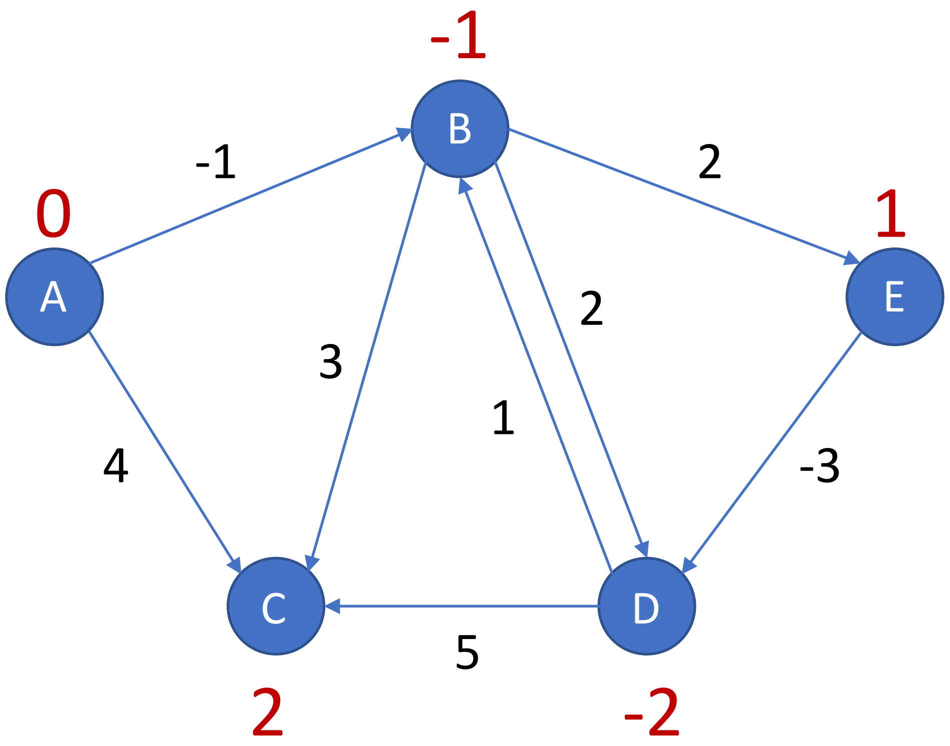


A	B	C	D	E
0	∞	∞	∞	∞
0	∞	∞	∞	∞
0	-1	∞	∞	∞
0	-1	4	∞	∞
0	-1	4	∞	∞
0	-1	4	∞	1
0	-1	4	1	1
0	-1	2	1	1
0	-1	2	-2	1

```
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*.

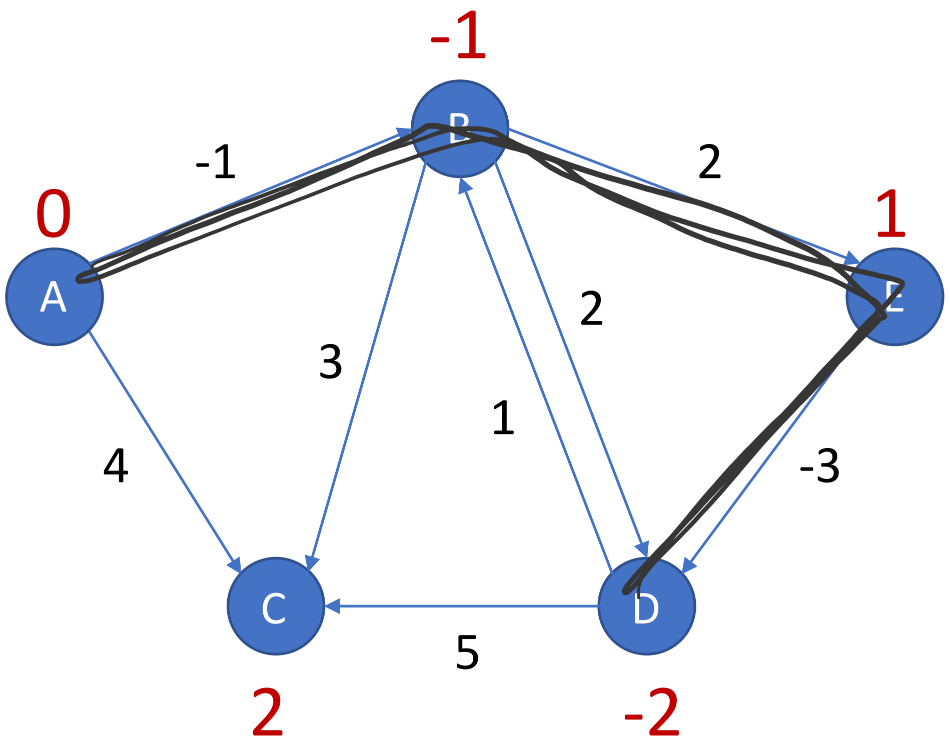
k=3:



A	B	C	D	E
0	∞	∞	∞	∞
0	∞	∞	∞	∞
0	-1	∞	∞	∞
0	-1	4	∞	∞
0	-1	4	∞	∞
0	-1	4	∞	1
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) = 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=4:



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

for k = n

$O(nm)$

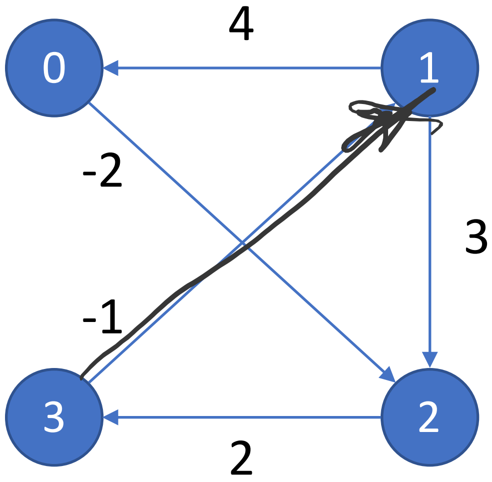
$m = n^2$

$O(n^3)$

Floyd-Warshall Algorithm Example

Initialization:

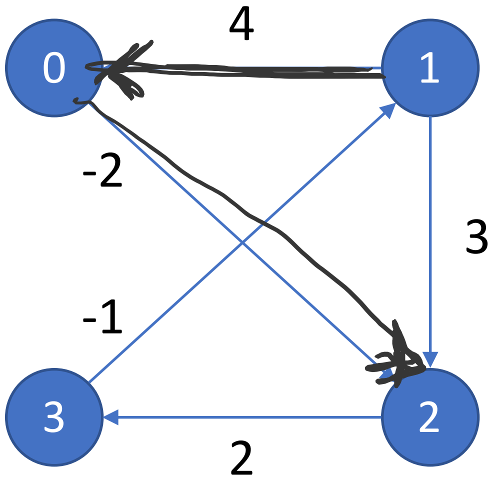
$$A^{l} = \begin{bmatrix} 0 & \infty & -2 & \infty \\ 4 & 0 & 3 & \infty \\ \infty & \infty & 0 & 2 \\ \infty & -1 & \infty & 0 \end{bmatrix}$$




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

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!")
```

← Base Case

k =0:

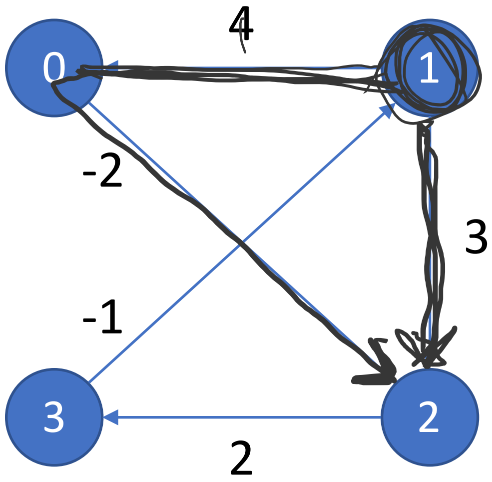


$$A^{l} = \begin{bmatrix} 0 & \infty & -2 & \infty \\ 4 & 0 & 3 & \infty \\ \infty & \infty & 0 & 2 \\ \infty & -1 & \infty & 0 \end{bmatrix}$$

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

```
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(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!")
```

k =0:



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

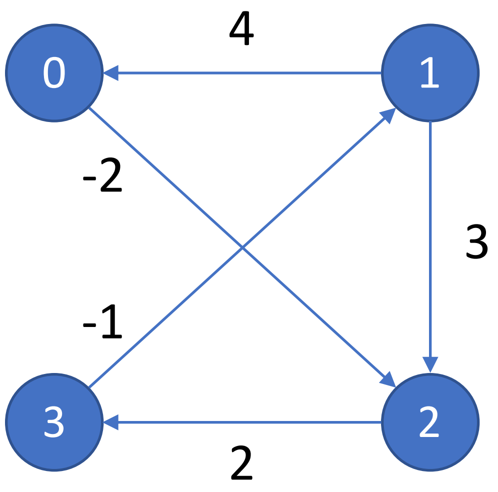
$$A^0 = \begin{bmatrix} 0 & \infty & -2 & \infty \\ 4 & 0 & 2 & \infty \\ \infty & \infty & 0 & 2 \\ \infty & -1 & \infty & 0 \end{bmatrix}$$

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

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

```
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(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!")
```

k =1:



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

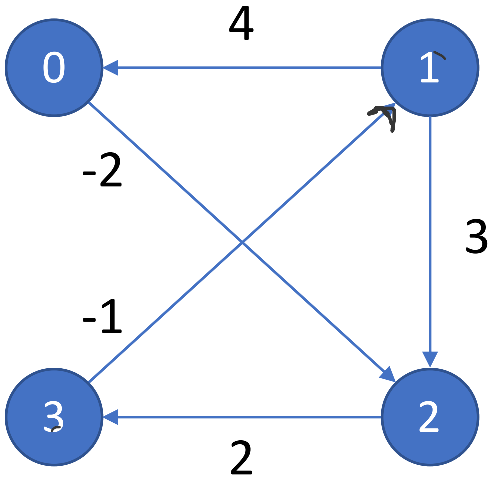
$$A^0 = \begin{bmatrix} 0 & \infty & -2 & \infty \\ 4 & 0 & \mathbf{2} & \infty \\ \infty & \infty & 0 & 2 \\ \infty & -1 & \infty & 0 \end{bmatrix}$$

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

```
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!")
```

k =1:

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

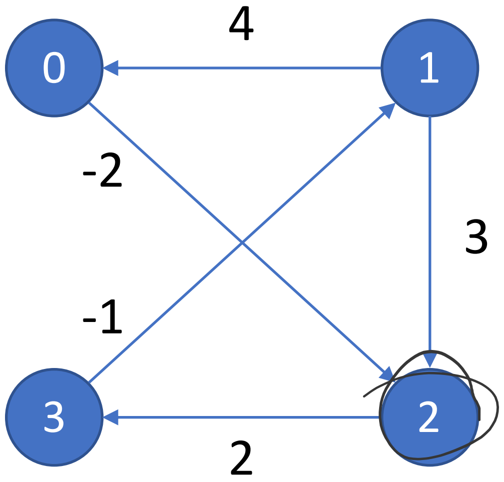


$$A^0 = \begin{bmatrix} 0 & \infty & -2 & \infty \\ 4 & 0 & \mathbf{2} & \infty \\ \infty & \infty & 0 & 2 \\ \infty & -1 & \infty & 0 \end{bmatrix} \qquad A^1 = \begin{bmatrix} 0 & \infty & -2 & \infty \\ 4 & 0 & 2 & \infty \\ \infty & \infty & 0 & 2 \\ \mathbf{3} & -1 & \mathbf{1} & 0 \end{bmatrix}$$

```
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!")
```

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

k =2:



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

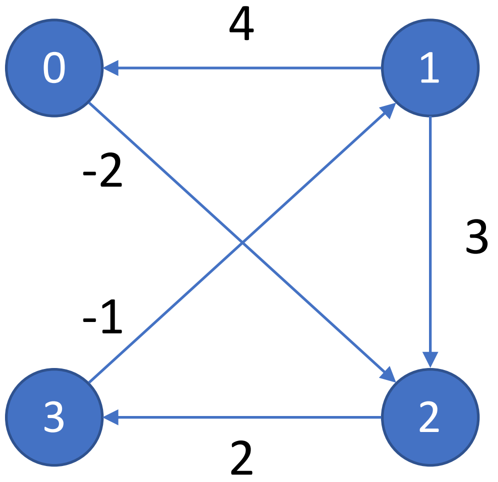
$$A^0 = \begin{bmatrix} 0 & \infty & -2 & \infty \\ 4 & 0 & \mathbf{2} & \infty \\ \infty & \infty & 0 & 2 \\ \infty & -1 & \infty & 0 \end{bmatrix}$$

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

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

```
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!")
```

k =3:




```
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!")
```

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

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

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

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

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