University of Nottingham
UK | CHINA | MALAYSIA

COMP2054 Tutorial Session 8: Floyd-Warshall Algorithm

Rebecca Tickle
Warren Jackson
AbdulHakim Ibrahim

Session outcomes

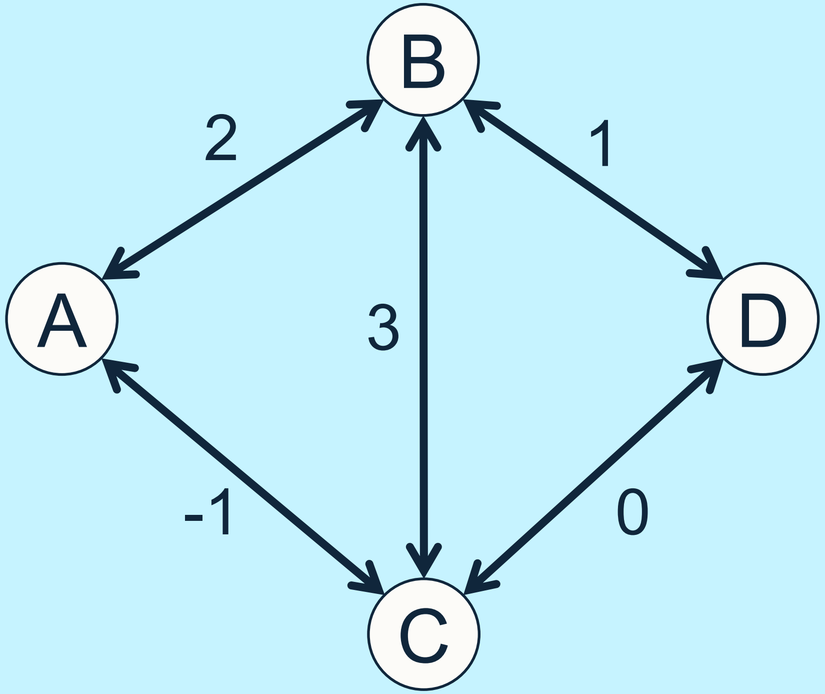
- Understand how to solve all-pairs shortest path problem using dynamic programming.
- Apply Floyd-Warshall to directed graphs to solve all-pairs shortest path problem.

University of Nottingham
UK | CHINA | MALAYSIA

All-Pairs Shortest Paths

All-pairs shortest paths problem

- Given a directed or undirected graph, find the shortest paths (costs) between all pairs of nodes.



Floyd-Warshall

Dynamic programming algorithm for all-pairs shortest paths

Floyd-Warshall algorithm

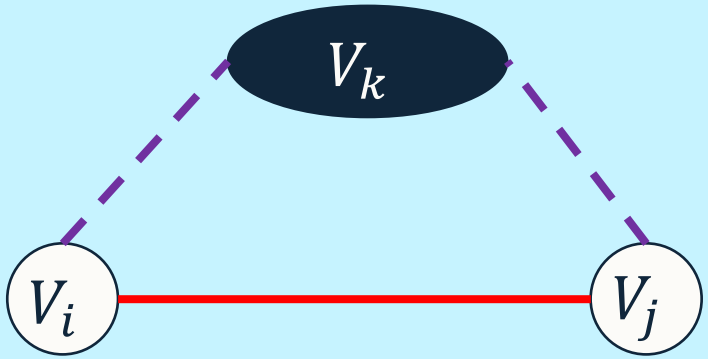
- Given a directed or undirected graph, find the shortest paths (costs) between all pairs of nodes.
- Uses **dynamic programming** to build up the graph from:
  - No intermediate nodes…
  - …to considering all nodes being allowed as intermediate nodes.

Important notations

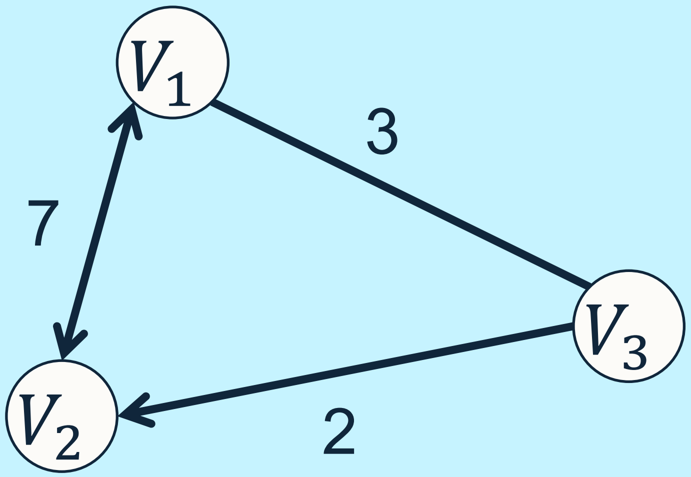
- $d(i, j, k)$ - the shortest distance between nodes i and j through some subset (including the empty set) of $\{V_1, \ldots, V_k\}$.

Important notations

- $d(i, j, k)$ - the shortest distance between nodes i and j through some subset (including the empty set) of $\{V_1, \dots, V_k\}$.
- $d(i, j, k) = \min[d(i, j, k-1),\ d(i, k, k-1) + d(k, j, k-1)]$



Floyd-Warshall Example: Initialisation



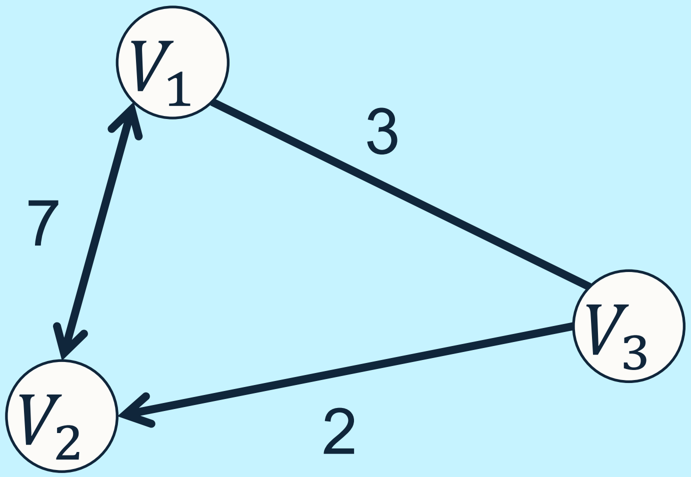
- **Initialise the adjacency matrix**

$i \backslash j$	V_1	V_2	V_3
V_1	0		
V_2		0	
V_3			0

- $d(i, j, 0)$
- Allowed intermediate nodes: {}

All $d(i, i) = 0$

Floyd-Warshall Example: Initialisation



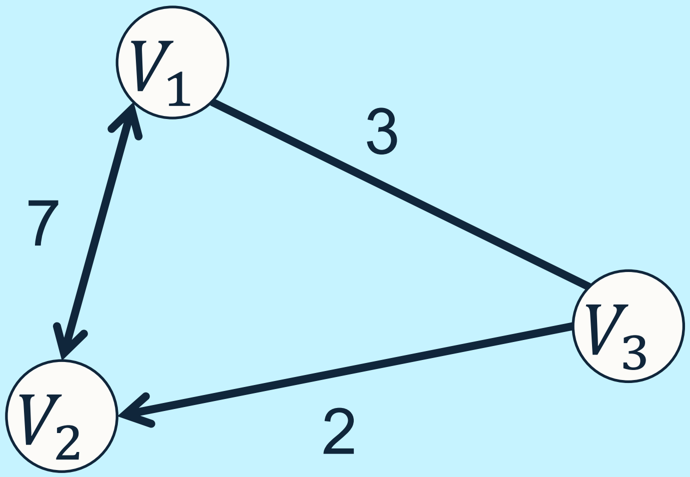
- **Initialise the adjacency matrix**

$i \backslash j$	V_1	V_2	V_3
V_1	0	7	3
V_2	7	0	
V_3	3	2	0

- $d(i, j, 0)$
- Allowed intermediate nodes: {}

If there is an edge linking two nodes, add the weight to the adjacency matrix.

Floyd-Warshall Example: Initialisation



- **Initialise the adjacency matrix**

$i \backslash j$	V_1	V_2	V_3
V_1	0	7	3
V_2	7	0	∞
V_3	3	2	0

- $d(i, j, 0)$
- Allowed intermediate nodes: {}

If there is no edge, and we cannot get from one node to another directly, we add ∞

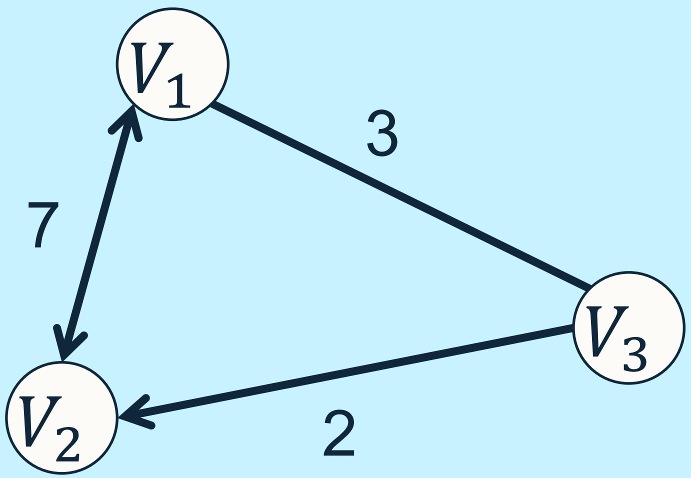
Floyd-Warshall Example

- Using the definition of:

$$d(i,j,k) = \min[\color{red}{d(i,j,k-1)}, \color{blue}{d(i,k,k-1) + d(k,j,k-1)}]$$

- Repeat for $k = 1$ to K (the number of vertices):

Insert V_k as an intermediate node and update the matrix

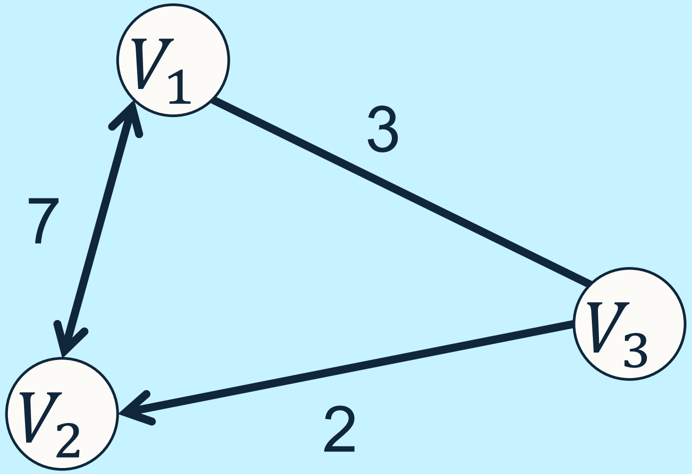


$i \backslash j$	V_1	V_2	V_3
V_1	0	7	3
V_2	7	0	∞
V_3	3	2	0

Floyd-Warshall Example: k=1

$$d(i,j,k) = \min[d(i,j,k-1), d(i,k,k-1) + d(k,j,k-1)]$$

- $k = 1$
- $d(i,j,1) = \min[d(i,j,0), d(i,1,0) + d(1,j,0)]$
- Intermediate nodes $= \{V_1\}$



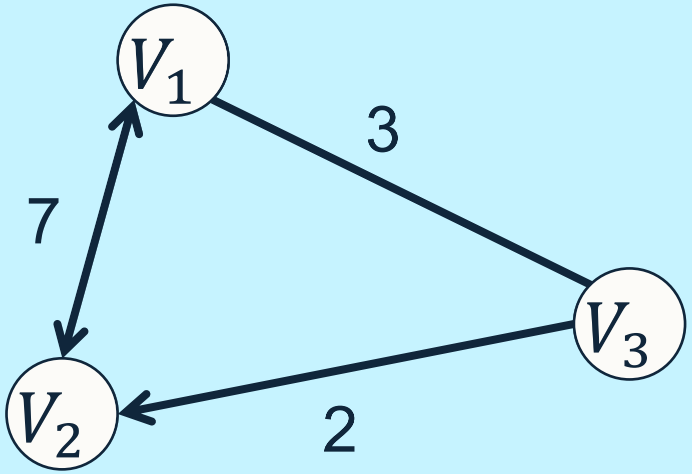
$k = 0$

$i \backslash j$	V_1	V_2	V_3
V_1	0	7	3
V_2	7	0	∞
V_3	3	2	0

Floyd-Warshall Example: k=1

$$d(i,j,k) = \min[d(i,j,k-1), d(i,k,k-1) + d(k,j,k-1)]$$

- $k = 1$
- $d(i,j,1) = \min[\boxed{d(i,j,0)}, d(i,1,0) + d(1,j,0)]$
- Intermediate nodes = $\{V_1\}$



$k = 0$

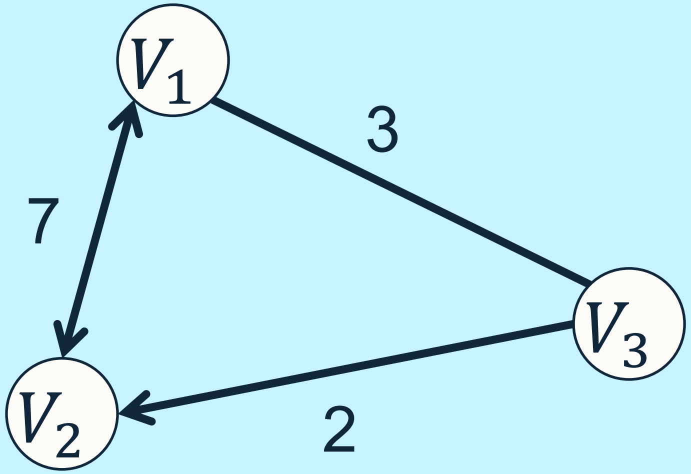
$i \backslash j$	V_1	V_2	V_3
V_1	0	7	3
V_2	7	0	∞
V_3	3	2	0

Floyd-Warshall Example: k=1

$$d(i,j,k) = \min[d(i,j,k-1), d(i,k,k-1) + d(k,j,k-1)]$$

- $k = 1$
- $d(i,j,1) = \min[\boxed{d(i,j,0)}, d(i,1,0) + d(1,j,0)]$
- Intermediate nodes $= \{V_1\}$

$$d(1,1,1) = \min[d(1,1,0), d(1,1,0) + d(1,1,0)] \\ = \min[0, 0+0] = 0$$



$k = 0$

i \ j	V_1	V_2	V_3
V_1	0	7	3
V_2	7	0	∞
V_3	3	2	0

$k = 1$

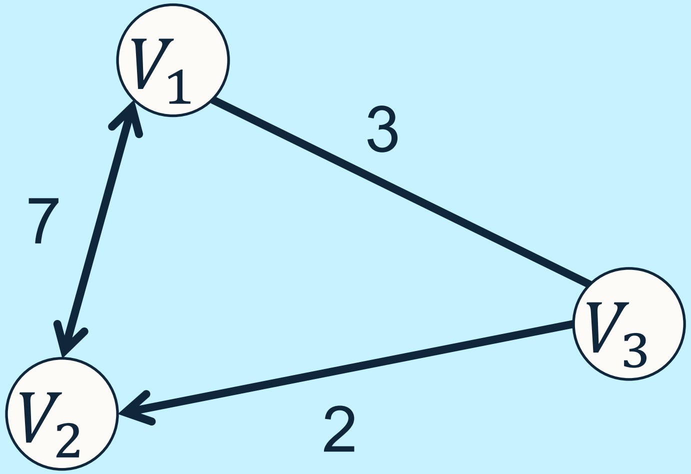
i \ j	V_1	V_2	V_3
V_1	0		
V_2			
V_3			

Floyd-Warshall Example: k=1

$$d(i,j,k) = \min[d(i,j,k{-}1), d(i,k,k{-}1) + d(k,j,k{-}1)]$$

- $k = 1$
- $d(i,j,1) = \min[\boxed{d(i,j,0)}, d(i,1,0) + d(1,j,0)]$
- Intermediate nodes $= \{V_1\}$

$$d(1,2,1) = \min[d(1,2,0), d(1,1,0) + d(1,2,0)]$$
$$= \min[7, 0+7] = 7$$



$k = 0$

$i \backslash j$	V_1	V_2	V_3
V_1	0	7	3
V_2	7	0	∞
V_3	3	2	0

$k = 1$

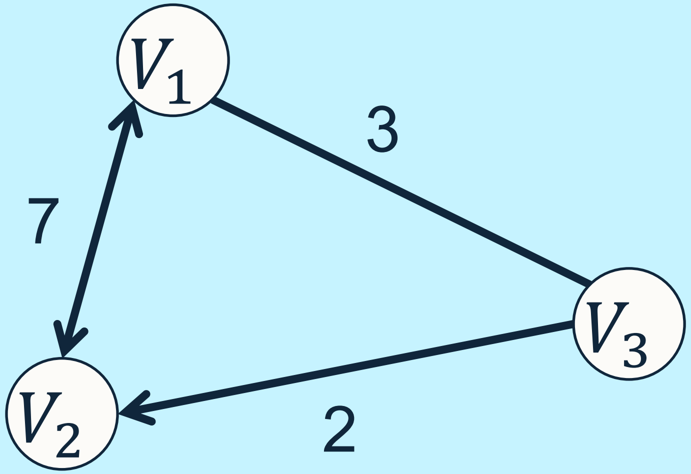
$i \backslash j$	V_1	V_2	V_3
V_1	0	7	
V_2			
V_3			

Floyd-Warshall Example: k=1

$$d(i,j,k) = \min[d(i,j,k-1), d(i,k,k-1) + d(k,j,k-1)]$$

- $k = 1$
- $d(i,j,1) = \min[\boxed{d(i,j,0)}, d(i,1,0) + d(1,j,0)]$
- Intermediate nodes = $\{V_1\}$

$$d(2,3,1) = \min[d(2,3,0), d(2,1,0) + d(1,3,0)]$$
$$= \min[\infty, 7 + 3] = 10$$



$k = 0$

$i \backslash j$	V_1	V_2	V_3
V_1	0	7	3
V_2	7	0	∞
V_3	3	2	0

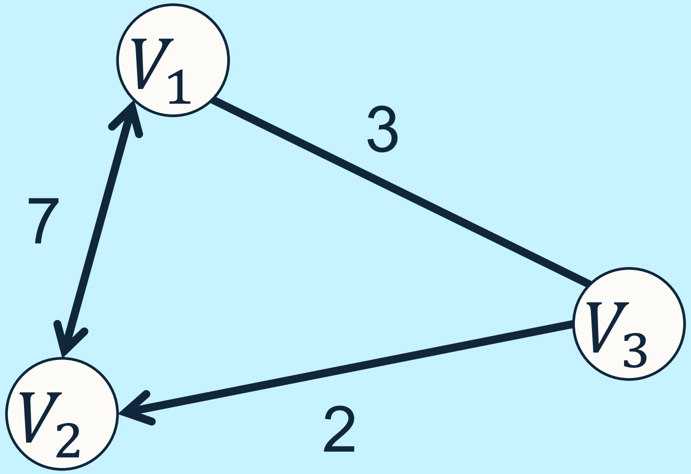
$k = 1$

$i \backslash j$	V_1	V_2	V_3
V_1	0	7	
V_2			10
V_3			

Floyd-Warshall Example: k=1

$$d(i,j,k) = \min[d(i,j,k-1), d(i,k,k-1) + d(k,j,k-1)]$$

- $k = 1$
- $d(i,j,1) = \min[\boxed{d(i,j,0)}, d(i,1,0) + d(1,j,0)]$
- Intermediate nodes = $\{V_1\}$



$k = 0$

$i \backslash j$	V_1	V_2	V_3
V_1	0	7	3
V_2	7	0	∞
V_3	3	2	0

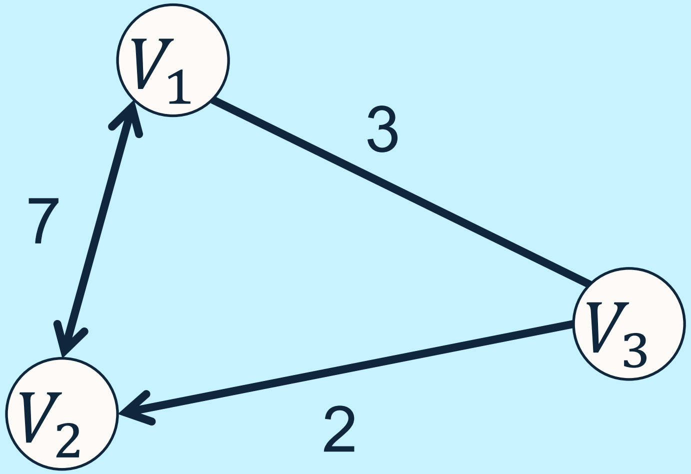
$k = 1$

$i \backslash j$	V_1	V_2	V_3
V_1	0	7	3
V_2	7	0	10
V_3	3	2	0

Floyd-Warshall Example

$$d(i,j,k) = \min[d(i,j,k-1), d(i,k,k-1) + d(k,j,k-1)]$$

- Working cell-by-cell in this way is quite laborious and error prone.
- There is a “shortcut” we can use to make the working more straightforward…



$k = 0$

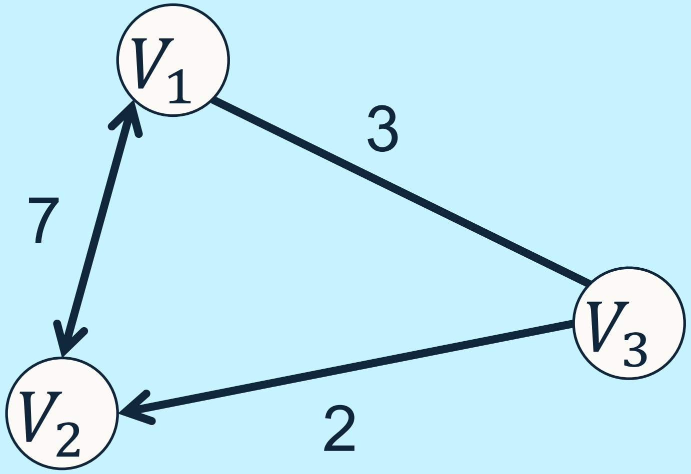
$i \backslash j$	V_1	V_2	V_3
V_1	0	7	3
V_2	7	0	∞
V_3	3	2	0

$k = 1$

$i \backslash j$	V_1	V_2	V_3
V_1	0	7	3
V_2	7	0	10
V_3	3	2	0

Floyd-Warshall Example: Shortcut

$d(i,j,k) = \min[d(i,j,k-1), \boldsymbol{d(i,k,k-1)} + \boldsymbol{d(k,j,k-1)}]$



- $k = 1$
- $d(i,j,1) = \min[d(i,j,0), \boldsymbol{d(i,1,0)} + \boldsymbol{d(1,j,0)}]$
- Intermediate nodes $= \{V_1\}$

$k = 0$

i \ j	V_1	V_2	V_3
V_1	0	7	3
V_2	7	0	∞
V_3	3	2	0

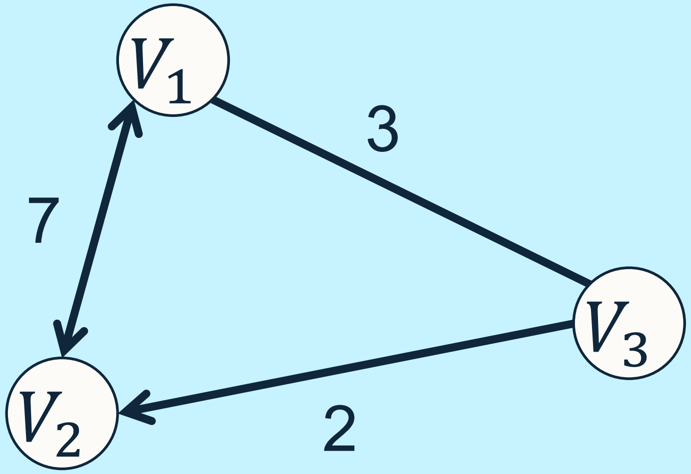
To calculate the cost matrix for $k = 1$, we want to consider $\{V_1\}$ as the intermediate node(s). Hence we use the row and column corresponding to V_1 (left).

$k = 1$

i \ j	V_1	V_2	V_3
V_1			
V_2			
V_3			

Floyd-Warshall Example: Shortcut

$d(i,j,k) = \min[d(i,j,k-1), \boldsymbol{d(i,k,k-1)} + \boldsymbol{d(k,j,k-1)}]$



- $k = 1$
- $d(i,j,1) = \min[d(i,j,0), \boldsymbol{d(i,1,0)} + \boldsymbol{d(1,j,0)}]$
- Intermediate nodes $= \{V_1\}$

$k = 0$

$i \backslash j$	V_1	V_2	V_3
V_1	0	7	3
V_2	7	0	∞
V_3	3	2	0

Intermediate sum

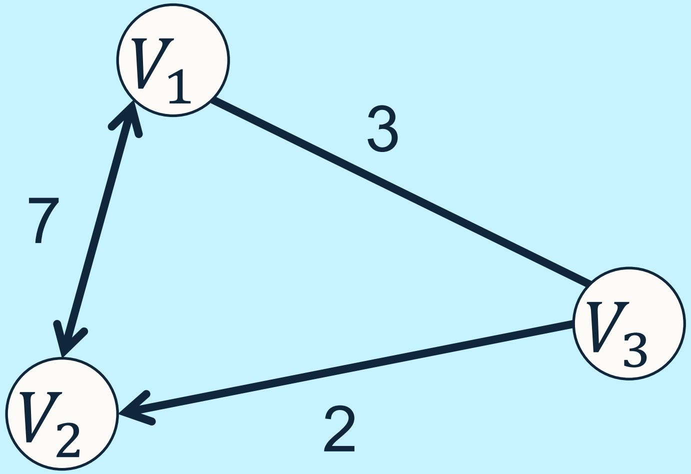
$i \backslash j$	V_1	V_2	V_3	
V_1				0
V_2				7
V_3				3
	0	7	3	

$k = 1$

$i \backslash j$	V_1	V_2	V_3
V_1			
V_2			
V_3			

Floyd-Warshall Example: Shortcut

$$d(i,j,k) = \min[d(i,j,k-1), \boldsymbol{d(i,k,k-1)} + \boldsymbol{d(k,j,k-1)}]$$



- $k = 1$
- $d(i,j,1) = \min[d(i,j,0), \boldsymbol{d(i,1,0)} + \boldsymbol{d(1,j,0)}]$
- Intermediate nodes = $\{V_1\}$

$k = 0$

$i \backslash j$	V_1	V_2	V_3
V_1	0	7	3
V_2	7	0	∞
V_3	3	2	0

Intermediate sum

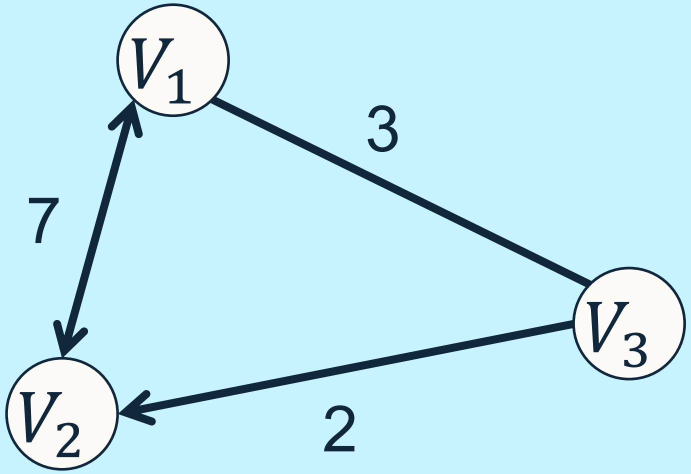
$i \backslash j$	V_1	V_2	V_3	
V_1	0+0	0+7	0+3	**0**
V_2	7+0	7+7	7+3	**7**
V_3	3+0	3+7	3+3	**3**
	0	**7**	**3**	

$k = 1$

$i \backslash j$	V_1	V_2	V_3
V_1			
V_2			
V_3			

Floyd-Warshall Example: Shortcut

$d(i,j,k) = \min[d(i,j,k-1), \mathbf{d(i,k,k-1)} + \mathbf{d(k,j,k-1)}]$



- $k = 1$
- $d(i,j,1) = \min[d(i,j,0), \mathbf{d(i,1,0)} + \mathbf{d(1,j,0)}]$
- Intermediate nodes $= \{V_1\}$

$k = 0$

$i \backslash j$	V_1	V_2	V_3
V_1	0	7	3
V_2	7	0	∞
V_3	3	2	0

Intermediate sum

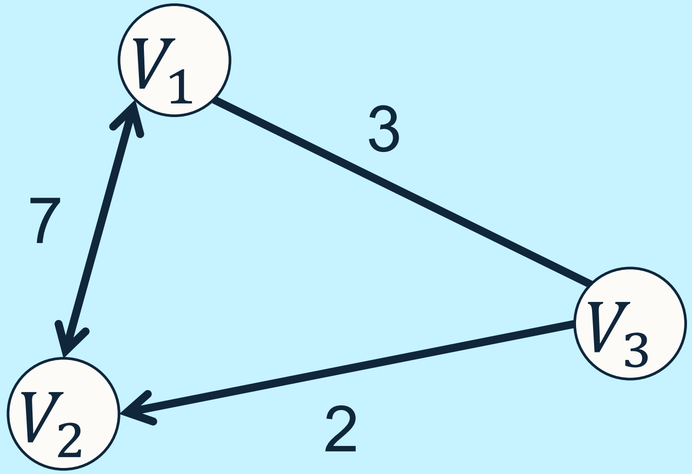
$i \backslash j$	V_1	V_2	V_3	
V_1	0	7	3	**0**
V_2	7	14	10	**7**
V_3	3	10	6	**3**
	0	**7**	**3**	

$k = 1$

$i \backslash j$	V_1	V_2	V_3
V_1			
V_2			
V_3			

Floyd-Warshall Example: Shortcut

$d(i,j,k) = \min[d(i,j,k-1), \boldsymbol{d(i,k,k-1)} + \boldsymbol{d(k,j,k-1)}]$



- $k = 1$
- $d(i,j,1) = \min[d(i,j,0), \boldsymbol{d(i,1,0)} + \boldsymbol{d(1,j,0)}]$
- Intermediate nodes $= \{V_1\}$

$k = 0$

$i \backslash j$	V_1	V_2	V_3
V_1	0	7	3
V_2	7	0	∞
V_3	3	2	0

Intermediate sum

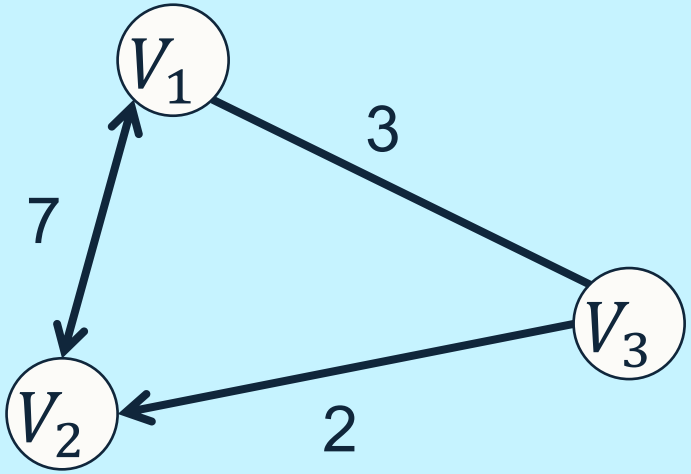
$i \backslash j$	V_1	V_2	V_3	
V_1	0	7	3	**0**
V_2	7	14	10	**7**
V_3	3	10	6	**3**
	0	**7**	**3**	

$k = 1$ as $\boldsymbol{min[k = 0, \text{Intermediate sum}]}$

$i \backslash j$	V_1	V_2	V_3
V_1	0	7	3
V_2	7	0	10
V_3	3	2	0

Floyd-Warshall Example: Shortcut

$d(i,j,k) = \min[d(i,j,k-1), \mathbf{d(i,k,k-1)} + \mathbf{d(k,j,k-1)}]$



- $k = 2$
- $d(i,j,2) = \min[d(i,j,1), \mathbf{d(i,2,1)} + \mathbf{d(2,j,1)}]$
- Intermediate nodes $= \{V_1, V_2\}$

$k = 1$

$i \backslash j$	V_1	V_2	V_3
V_1	0	7	3
V_2	7	0	10
V_3	3	2	0

Intermediate sum

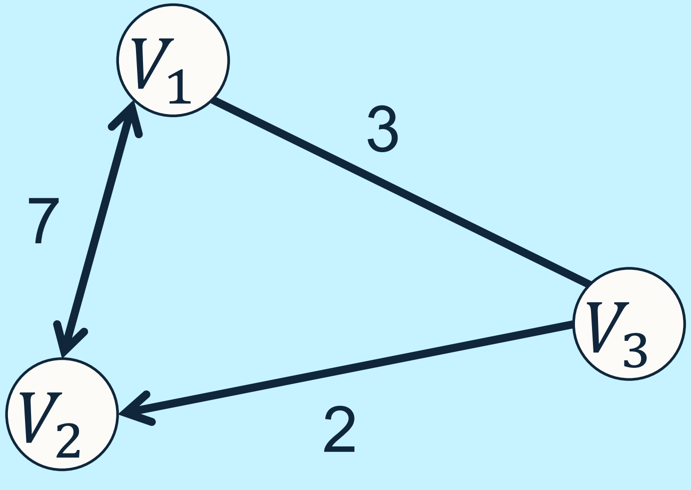
$i \backslash j$	V_1	V_2	V_3	
V_1				7
V_2				0
V_3				2
	7	0	10	

$k = 2$ **as** $\boldsymbol{min}[\boldsymbol{k} = \mathbf{1}, \textbf{Intermediate sum}]$

$i \backslash j$	V_1	V_2	V_3
V_1			
V_2			
V_3			

Floyd-Warshall Example: Shortcut

$$d(i,j,k) = \min[d(i,j,k{-}1), \mathbf{d(i,k,k{-}1)} + \mathbf{d(k,j,k{-}1)}]$$



- $k = 2$
- $d(i,j,2) = \min[d(i,j,1), \mathbf{d(i,2,1)} + \mathbf{d(2,j,1)}]$
- Intermediate nodes $= \{V_1, V_2\}$

$k = 1$

$i \backslash j$	V_1	V_2	V_3
V_1	0	7	3
V_2	7	0	10
V_3	3	2	0

Intermediate sum

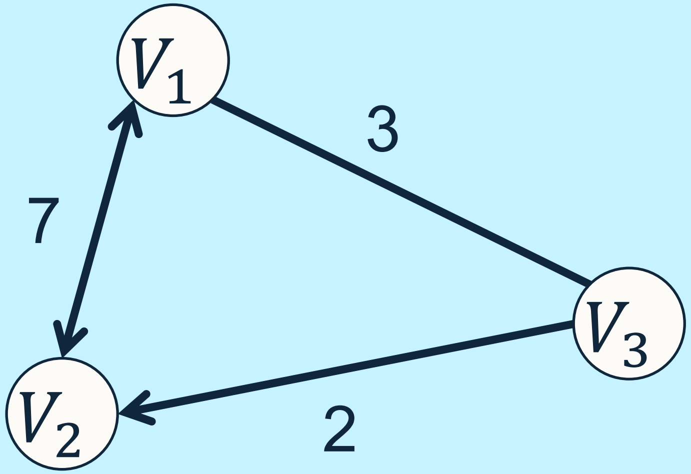
$i \backslash j$	V_1	V_2	V_3	
V_1	14	7	17	**7**
V_2	7	0	10	**0**
V_3	9	2	12	**2**
	7	**0**	**10**	

$k = 2$ as $min[\mathbf{k = 1}, \textbf{Intermediate sum}]$

$i \backslash j$	V_1	V_2	V_3
V_1			
V_2			
V_3			

Floyd-Warshall Example: Shortcut

$d(i,j,k) = \min[d(i,j,k-1), \boldsymbol{d(i,k,k-1)} + \boldsymbol{d(k,j,k-1)}]$



- $k = 2$
- $d(i,j,2) = \min[d(i,j,1), \boldsymbol{d(i,2,1)} + \boldsymbol{d(2,j,1)}]$
- Intermediate nodes = $\{V_1, V_2\}$

$k = 1$

$i \backslash j$	V_1	V_2	V_3
V_1	0	7	3
V_2	7	0	10
V_3	3	2	0

Intermediate sum

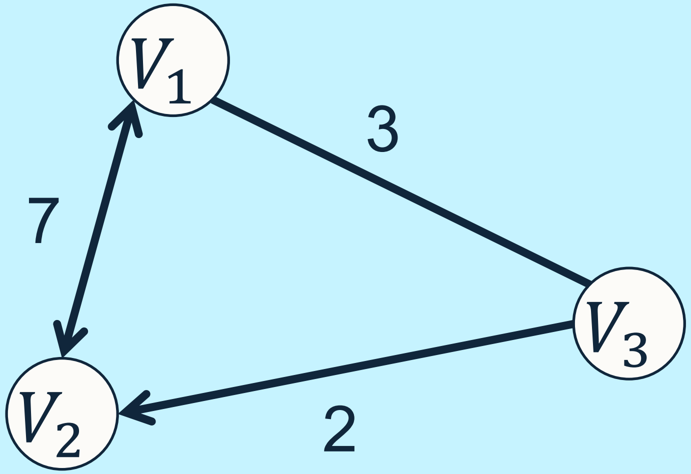
$i \backslash j$	V_1	V_2	V_3	
V_1	14	7	17	**7**
V_2	7	0	10	**0**
V_3	9	2	12	**2**
	7	**0**	**10**	

$k = 2$ as $\boldsymbol{min[k = 1, \text{Intermediate sum}]}$

$i \backslash j$	V_1	V_2	V_3
V_1	0	7	3
V_2	7	0	10
V_3	3	2	0

Floyd-Warshall Example: Shortcut

$d(i,j,k) = \min[d(i,j,k-1), \mathbf{d(i,k,k-1)} + \mathbf{d(k,j,k-1)}]$



- $k = 3$
- $d(i,j,3) = \min[d(i,j,2), \mathbf{d(i,3,2)} + \mathbf{d(3,j,2)}]$
- Intermediate nodes $= \{V_1, V_2, V_3\}$

$k = 2$

$i \backslash j$	V_1	V_2	V_3
V_1	0	7	3
V_2	7	0	10
V_3	3	2	0

Intermediate sum

$i \backslash j$	V_1	V_2	V_3	
V_1	6	5	3	**3**
V_2	13	12	10	**10**
V_3	3	2	0	**0**
	3	**2**	**0**	

$k = 3$ as $min[\mathbf{k = 2}, \textbf{Intermediate sum}]$

$i \backslash j$	V_1	V_2	V_3
V_1	0	5	3
V_2	7	0	10
V_3	3	2	0

Floyd-Warshall Example: Complete Shortcut

Intermediate sum:

$i \backslash j$	V_1	V_2	V_3
V_1	0	7	3

$i \backslash j$	V_1	V_2	V_3
V_1	14	7	17

$i \backslash j$	V_1	V_2	V_3
V_1	6	5	3
[illegible]	[illegible]	12	10
[illegible]	[illegible]	2	0

This slide demonstrates how each step links together from considering no intermediate nodes ($k = 0$) through to all nodes as intermediate nodes ($k = 3$).

[Advance slide to see all matrices]

$k = 0$

$i \backslash j$	V_1	V_2	V_3
V_1	0	7	3
V_2	7	0	∞
V_3	3	2	0

$i \backslash j$	V_1	V_2	V_3
V_1	0	7	3
V_2	7	0	10
V_3	3	2	0

$i \backslash j$	V_1	V_2	V_3
V_1	0	7	3
V_2	7	0	10
V_3	3	2	0

$i \backslash j$	V_1	V_2	V_3
V_1	0	5	3
V_2	7	0	10
V_3	3	2	0

Floyd-Warshall Example: Complete Shortcut

Intermediate sum:

$i \backslash j$	V_1	V_2	V_3
V_1	0	7	3
V_2	7	14	10
V_3	3	10	6

$i \backslash j$	V_1	V_2	V_3
V_1	14	7	17
V_2	7	0	10
V_3	9	2	12

$i \backslash j$	V_1	V_2	V_3
V_1	6	5	3
V_2	13	12	10
V_3	3	2	0

$k = 0$

$i \backslash j$	V_1	V_2	V_3
V_1	0	7	3
V_2	7	0	∞
V_3	3	2	0

$k = 1$

$i \backslash j$	V_1	V_2	V_3
V_1	0	7	3
V_2	7	0	10
V_3	3	2	0

$k = 2$

$i \backslash j$	V_1	V_2	V_3
V_1	0	7	3
V_2	7	0	10
V_3	3	2	0

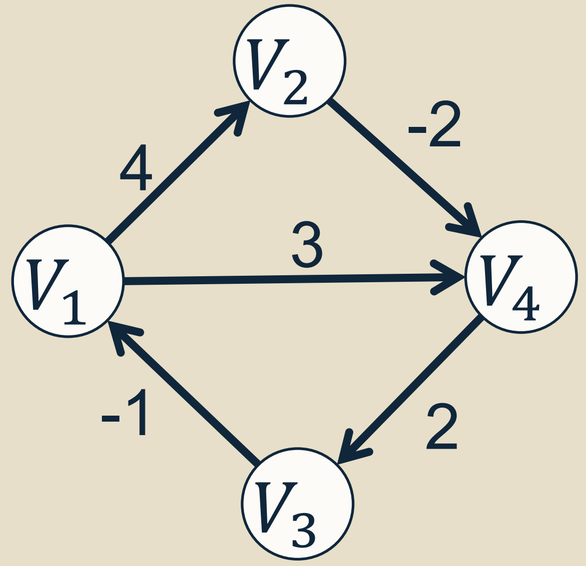
$k = 3$

$i \backslash j$	V_1	V_2	V_3
V_1	0	5	3
V_2	7	0	10
V_3	3	2	0

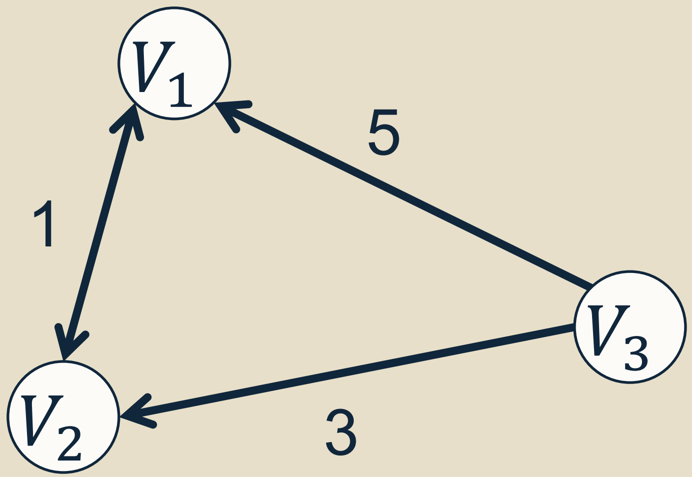
Floyd-Warshall Questions

Use the Floyd-Warshall algorithm to find the matrix of all-pairs shortest paths for the graphs below.

Q1.



Q2.



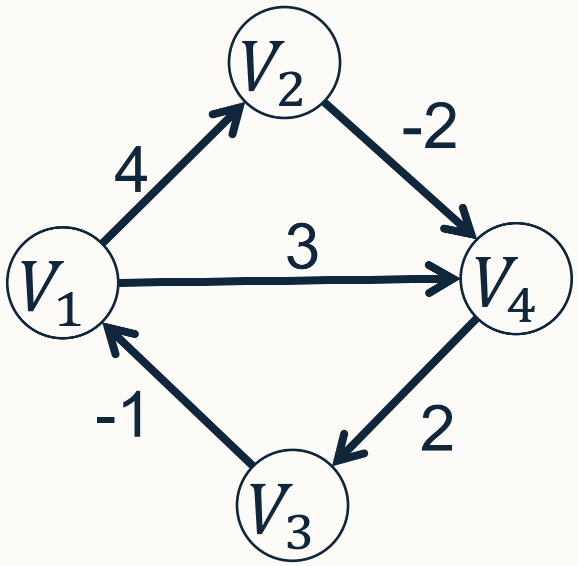
Answers – Q1

This graph has 4 vertices so we should construct a 4x4 matrix using the "shortcut" method. First initialise with the costs of direct edges from each node to each other node.

$$d(i,i,0) = 0$$

$k = 0$

$i \backslash j$	V_1	V_2	V_3	V_4
V_1	0			
V_2		0		
V_3			0	
V_4				0



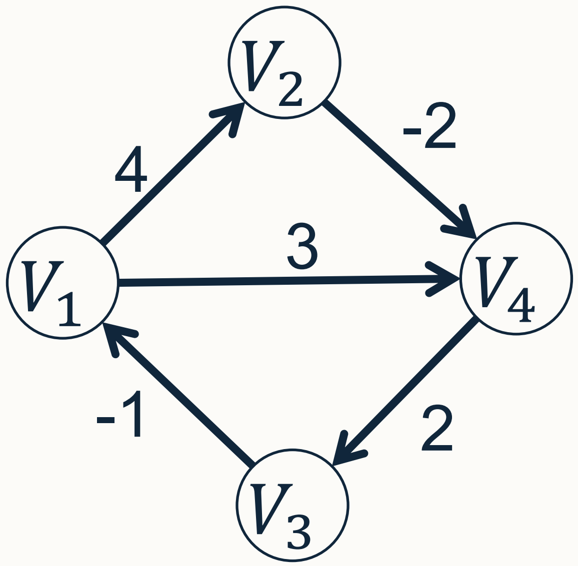
Answers – Q1

This graph has 4 vertices so we should construct a 4x4 matrix using the “shortcut” method. First initialise with the costs of direct edges from each node to each other node.

Enter all direct edges

$k = 0$

$i \backslash j$	V_1	V_2	V_3	V_4
V_1	0	4		3
V_2		0		-2
V_3	-1		0	
V_4			2	0



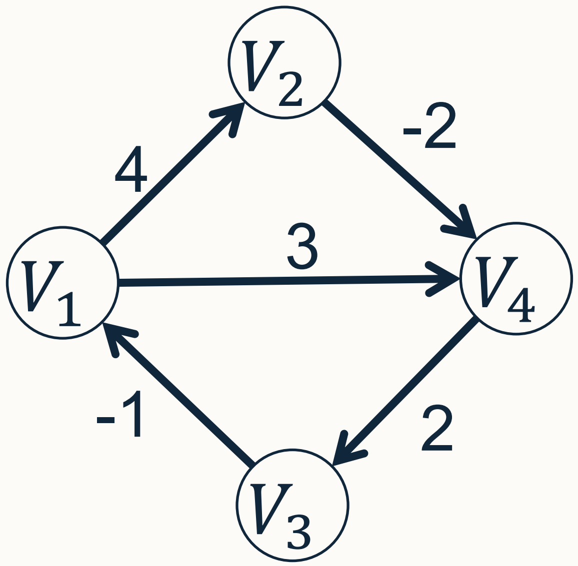
Answers – Q1

This graph has 4 vertices so we should construct a 4x4 matrix using the “shortcut” method. First initialise with the costs of direct edges from each node to each other node.

…and fill in the blanks (no direct edges) with infinity.

$k = 0$

$i \backslash j$	V_1	V_2	V_3	V_4
V_1	0	4	∞	3
V_2	∞	0	∞	-2
V_3	-1	∞	0	∞
V_4	∞	∞	2	0



Answers – Q1

$k = 1$, calculate intermediate sum for $\{V_1\}$ and take min of $k = 0$ and intermediate sum to construct matrix for $k = 1$.

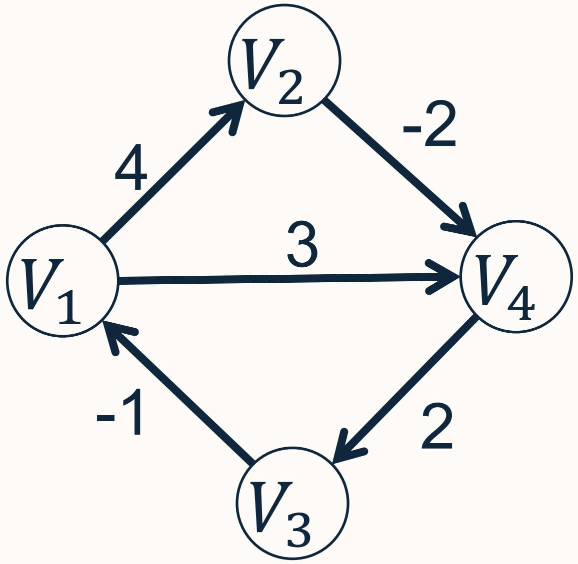
$i \backslash j$	0	4	∞	3
0	0	4	∞	3
∞	∞	∞	∞	∞
-1	-1	3	∞	2
∞	∞	∞	∞	∞

$k = 0$

$i \backslash j$	V_1	V_2	V_3	V_4
V_1	0	4	∞	3
V_2	∞	0	∞	-2
V_3	-1	∞	0	∞
V_4	∞	∞	2	0

$k = 1$

$i \backslash j$	V_1	V_2	V_3	V_4
V_1	0	4	∞	3
V_2	∞	0	∞	-2
V_3	-1	3	0	2
V_4	∞	∞	2	0



Answers – Q1

$k = 2$, calculate intermediate sum for $\{V_1, V_2\}$ and take min of $k = 1$ and intermediate sum to construct matrix for $k = 2$.

$i \backslash j$	0	4	∞	3
0	0	4	∞	3
∞	∞	∞	∞	∞
-1	-1	3	∞	2
∞	∞	∞	∞	∞

$i \backslash j$	∞	0	∞	-2
4	∞	4	∞	2
0	∞	0	∞	-2
3	∞	3	∞	1
∞	∞	∞	∞	∞

$k = 0$

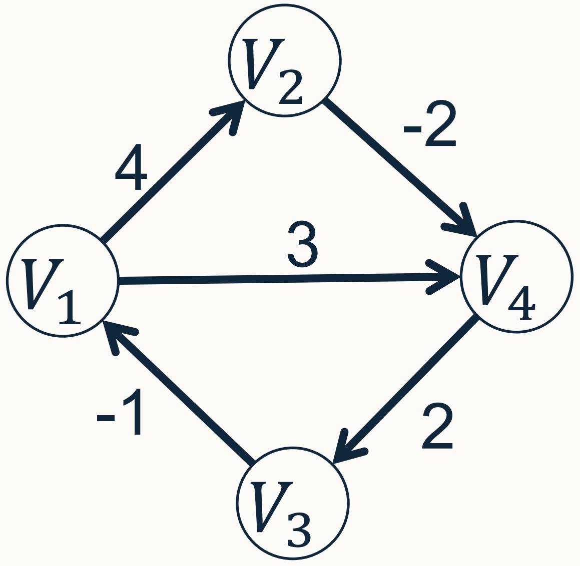
$i \backslash j$	V_1	V_2	V_3	V_4
V_1	0	4	∞	3
V_2	∞	0	∞	-2
V_3	-1	∞	0	∞
V_4	∞	∞	2	0

$k = 1$

$i \backslash j$	V_1	V_2	V_3	V_4
V_1	0	4	∞	3
V_2	∞	0	∞	-2
V_3	-1	3	0	2
V_4	∞	∞	2	0

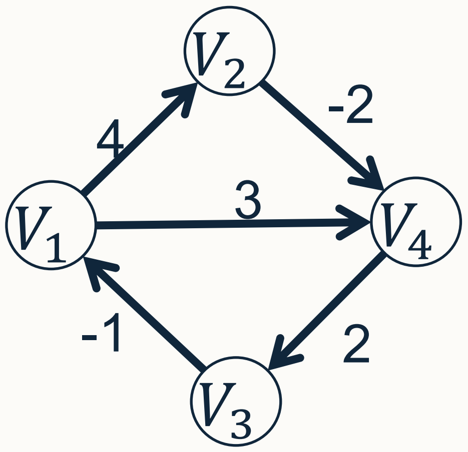
$k = 2$

$i \backslash j$	V_1	V_2	V_3	V_4
V_1	0	4	∞	2
V_2	∞	0	∞	-2
V_3	-1	3	0	1
V_4	∞	∞	2	0



Answers – Q1

$k = 3$, calculate intermediate sum for $\{V_1, V_2, V_3\}$ and take min of $k = 2$ and intermediate sum to construct matrix for $k = 3$.



$i \backslash j$	0	4	∞	3
0	0	4	∞	3
∞	∞	∞	∞	∞
-1	-1	3	∞	2
∞	∞	∞	∞	∞

$i \backslash j$	∞	0	∞	-2
4	∞	4	∞	2
0	∞	0	∞	-2
3	∞	3	∞	1
∞	∞	∞	∞	∞

$i \backslash j$	-1	3	0	1
∞	∞	∞	∞	∞
∞	∞	∞	∞	∞
0	-1	3	0	1
2	1	5	2	3

$k = 0$

$i \backslash j$	V_1	V_2	V_3	V_4
V_1	0	4	∞	3
V_2	∞	0	∞	-2
V_3	-1	∞	0	∞
V_4	∞	∞	2	0

$k = 1$

$i \backslash j$	V_1	V_2	V_3	V_4
V_1	0	4	∞	3
V_2	∞	0	∞	-2
V_3	-1	3	0	2
V_4	∞	∞	2	0

$k = 2$

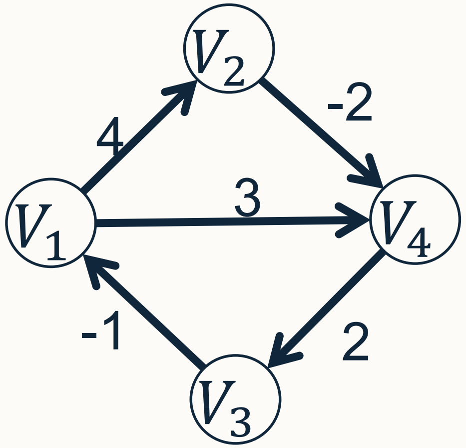
$i \backslash j$	V_1	V_2	V_3	V_4
V_1	0	4	∞	2
V_2	∞	0	∞	-2
V_3	-1	3	0	1
V_4	∞	∞	2	0

$k = 3$

$i \backslash j$	V_1	V_2	V_3	V_4
V_1	0	4	∞	2
V_2	∞	0	∞	-2
V_3	-1	3	0	1
V_4	1	5	2	0

Answers – Q1

$k = 4$, calculate intermediate sum for $\{V_1, V_2, V_3, V_4\}$ and take min of $k = 3$ and intermediate sum to construct matrix for $k = 4$. This matrix is the final matrix which gives the shortest path of all pairs of nodes in the graph.



$i \backslash j$	0	4	∞	3
0	0	4	∞	3
∞	∞	∞	∞	∞
-1	-1	3	∞	2
∞	∞	∞	∞	∞

$i \backslash j$	∞	0	∞	-2
4	∞	4	∞	2
0	∞	0	∞	-2
3	∞	3	∞	1
∞	∞	∞	∞	∞

$i \backslash j$	-1	3	0	1
∞	∞	∞	∞	∞
∞	∞	∞	∞	∞
0	-1	3	0	1
2	1	5	2	3

$i \backslash j$	1	5	2	0
2	3	7	4	2
-2	-1	3	0	-2
1	2	6	3	1
0	1	5	2	0

$k = 0$

$i \backslash j$	V_1	V_2	V_3	V_4
V_1	0	4	∞	3
V_2	∞	0	∞	-2
V_3	-1	∞	0	∞
V_4	∞	∞	2	0

$k = 1$

$i \backslash j$	V_1	V_2	V_3	V_4
V_1	0	4	∞	3
V_2	∞	0	∞	-2
V_3	-1	3	0	2
V_4	∞	∞	2	0

$k = 2$

$i \backslash j$	V_1	V_2	V_3	V_4
V_1	0	4	∞	2
V_2	∞	0	∞	-2
V_3	-1	3	0	1
V_4	∞	∞	2	0

$k = 3$

$i \backslash j$	V_1	V_2	V_3	V_4
V_1	0	4	∞	2
V_2	∞	0	∞	-2
V_3	-1	3	0	1
V_4	1	5	2	0

$k = 4$

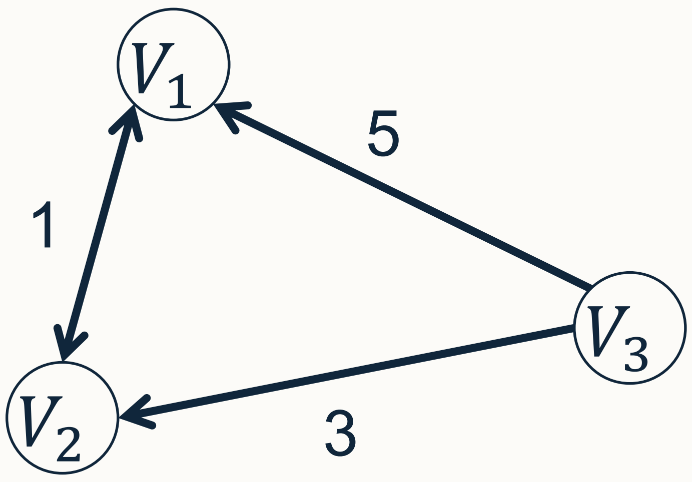
$i \backslash j$	V_1	V_2	V_3	V_4
V_1	0	4	4	2
V_2	-1	0	0	-2
V_3	-1	3	0	1
V_4	1	5	2	0

Answers – Q2

$k = 0$, calculate the initial matrix for direct edges only.

$k = 0$

$i \backslash j$	V_1	V_2	V_3
V_1	0	1	∞
V_2	1	0	∞
V_3	5	3	0



Answers – Q2

$k = 1$, calculate intermediate sum for $\{V_1\}$ and take min of $k = 0$ and intermediate sum to construct matrix for $k = 1$.

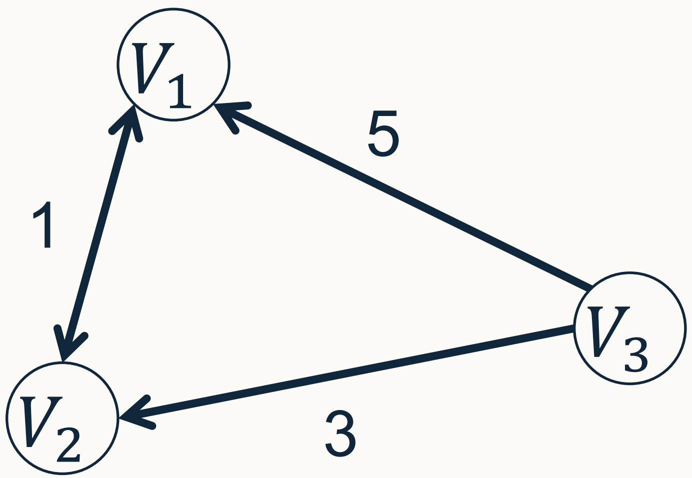
$i \backslash j$	0	1	∞
0	0	1	∞
1	1	2	∞
5	5	6	∞

$k = 0$

$i \backslash j$	V_1	V_2	V_3
V_1	0	1	∞
V_2	1	0	∞
V_3	5	3	0

$k = 1$

$i \backslash j$	V_1	V_2	V_3
V_1	0	1	∞
V_2	1	0	∞
V_3	5	3	0



Answers – Q2

$k = 2$, calculate intermediate sum for $\{V_1, V_2\}$ and take min of $k = 1$ and intermediate sum to construct matrix for $k = 2$.

$i \backslash j$	0	1	∞
0	0	1	∞
1	1	2	∞
5	5	6	∞

$i \backslash j$	1	0	∞
1	2	1	∞
0	1	0	∞
3	4	3	∞

$k = 0$

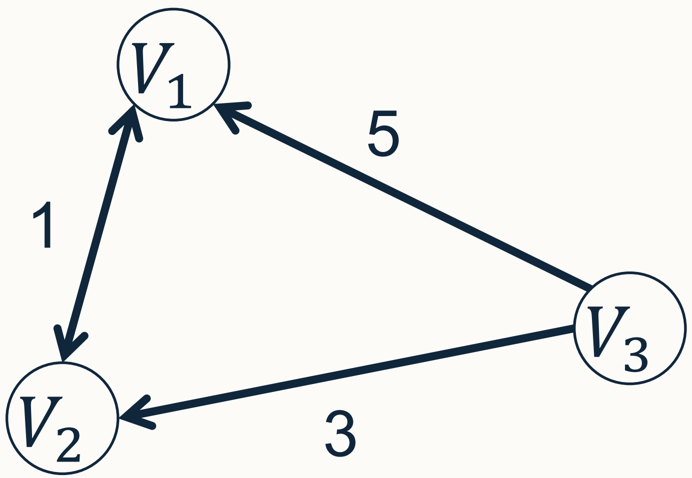
$i \backslash j$	V_1	V_2	V_3
V_1	0	1	∞
V_2	1	0	∞
V_3	5	3	0

$k = 1$

$i \backslash j$	V_1	V_2	V_3
V_1	0	1	∞
V_2	1	0	∞
V_3	5	3	0

$k = 2$

$i \backslash j$	V_1	V_2	V_3
V_1	0	1	∞
V_2	1	0	∞
V_3	4	3	0



Answers – Q2

$k = 3$, calculate intermediate sum for $\{V_1, V_2, V_3\}$ and take min of $k = 2$ and intermediate sum to construct matrix for $k = 3$. This matrix is the final matrix which gives the shortest path of all pairs of nodes in the graph.

$i \backslash j$	0	1	∞
0	0	1	∞
1	1	2	∞
5	5	6	∞

$i \backslash j$	1	0	∞
1	2	1	∞
0	1	0	∞
3	4	3	∞

$i \backslash j$	4	3	0
∞	∞	∞	∞
∞	∞	∞	∞
0	4	3	0

$k = 0$

$i \backslash j$	V_1	V_2	V_3
V_1	0	1	∞
V_2	1	0	∞
V_3	5	3	0

$k = 1$

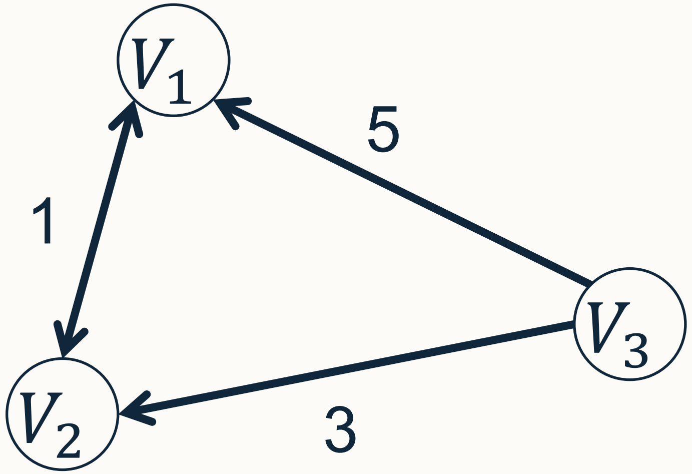
$i \backslash j$	V_1	V_2	V_3
V_1	0	1	∞
V_2	1	0	∞
V_3	5	3	0

$k = 2$

$i \backslash j$	V_1	V_2	V_3
V_1	0	1	∞
V_2	1	0	∞
V_3	4	3	0

$k = 3$

$i \backslash j$	V_1	V_2	V_3
V_1	0	1	∞
V_2	1	0	∞
V_3	4	3	0



Notice how there are still ∞ in
The final matrix. This represents
The fact there are no paths to V_3.

University of Nottingham
UK | CHINA | MALAYSIA

Thank you