University of Nottingham
UK | CHINA | MALAYSIA

COMP2054 Tutorial Session 8: Floyd-Warshall Algorithm

Rebecca Tickle
Warren Jackson
AbdulHakim Ibrahim

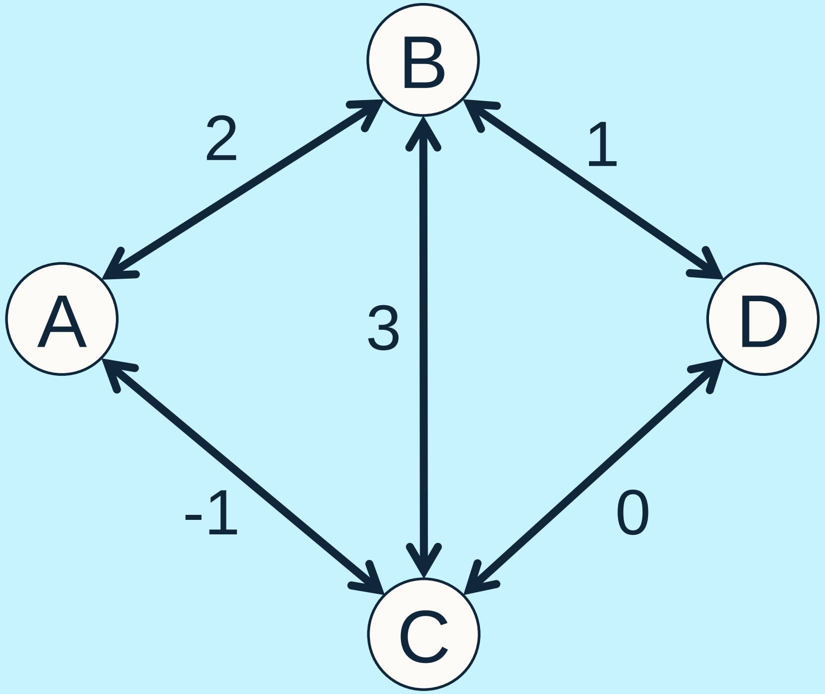
Session outcomes

- 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.

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

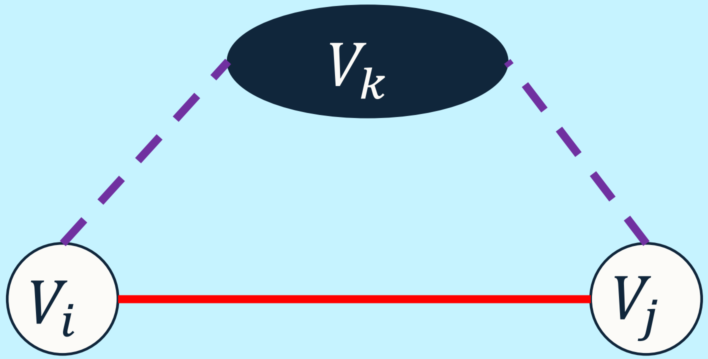
- Given a directed or undirected graph, find the shortest paths (costs) between all pairs of nodes.
- Uses **dynamic programming** to build up the graph from:
  - No intermediate nodes…
  - …to considering all nodes being allowed as intermediate nodes.

Important notations

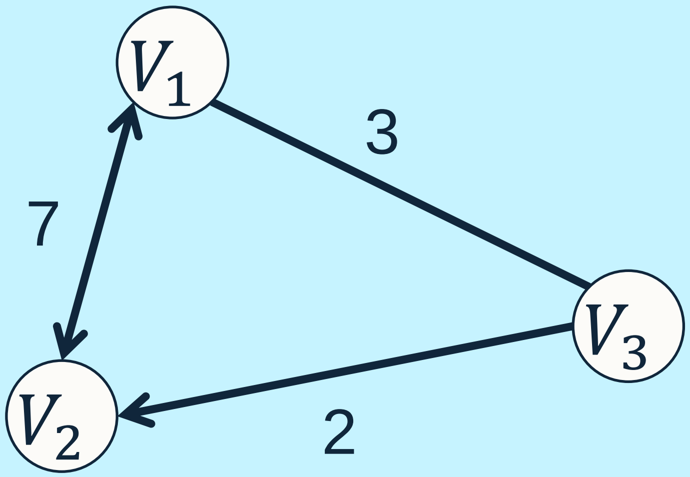
- $d(i, j, k)$ - the shortest distance between nodes i and j through some subset (including the empty set) of $\{V_1, \ldots, V_k\}$.

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



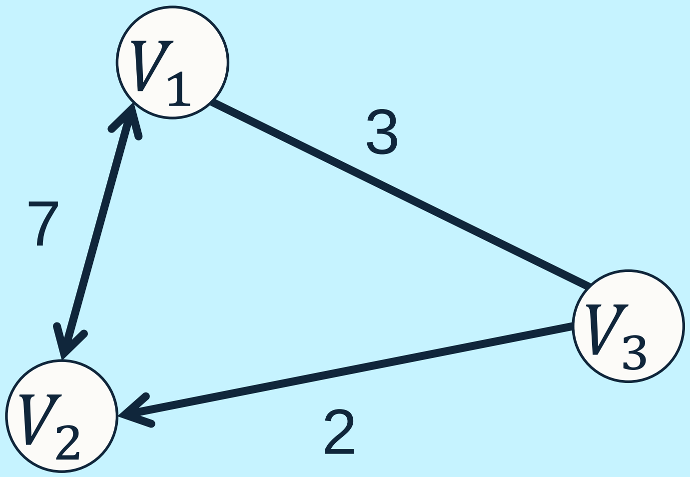
- **Initialise the adjacency matrix**

$i \backslash j$	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



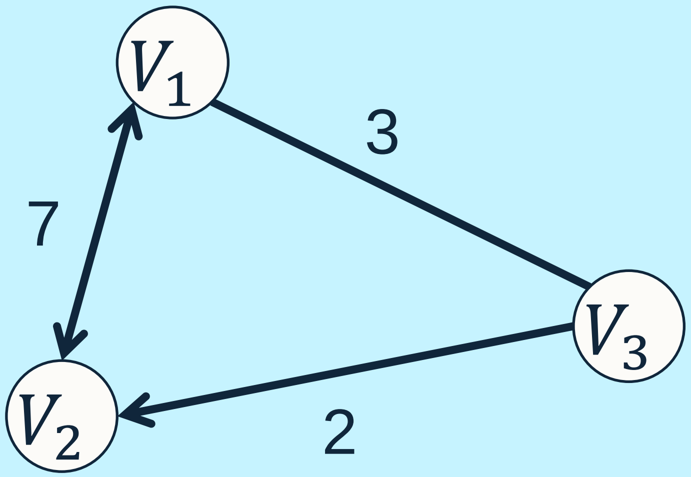
- **Initialise the adjacency matrix**

$i \backslash j$	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 a (directed) edge linking two nodes, add the weight to the adjacency matrix.

Floyd-Warshall Example: Initialisation



- **Initialise the adjacency matrix**

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

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

If there is no directed edge, add ∞

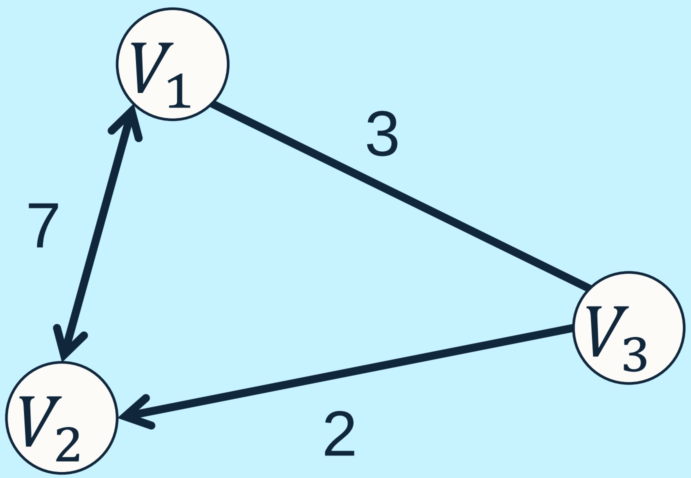
Floyd-Warshall Example

- Using the definition of:

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

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

Insert V_k as an intermediate node and update the matrix

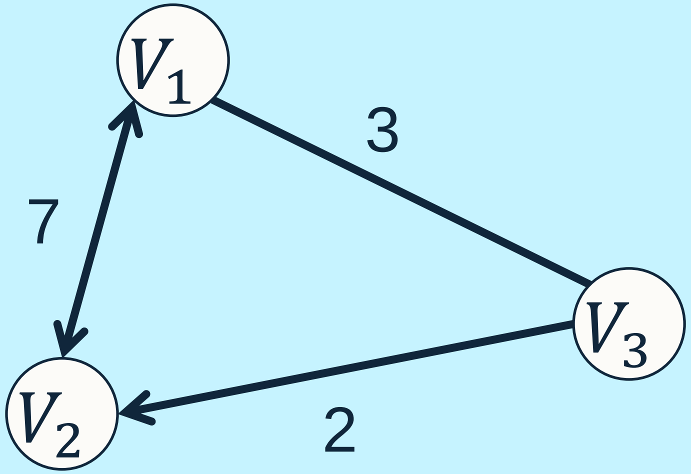


$i \backslash j$	V_1	V_2	V_3
V_1	0	7	3
V_2	7	0	∞
V_3	3	2	0

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 $= \{V_1\}$



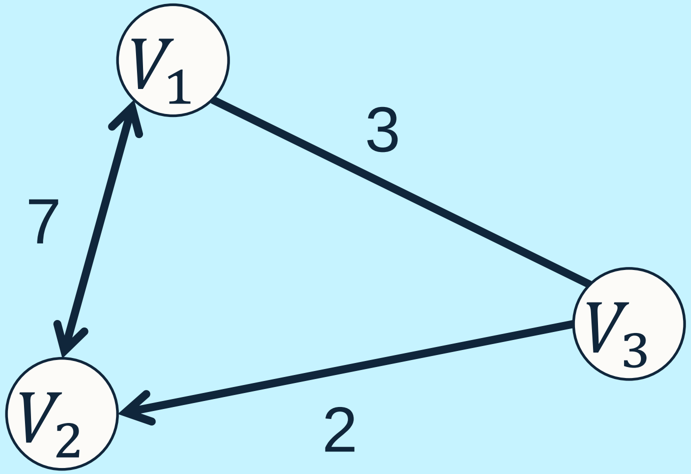
$k = 0$

$i \backslash j$	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 = $\{V_1\}$



$k = 0$

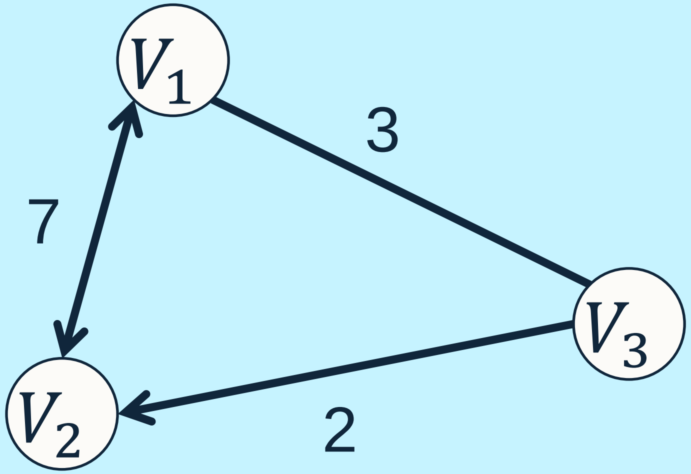
$i \backslash j$	V_1	V_2	V_3
V_1	0	7	3
V_2	7	0	∞
V_3	3	2	0

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 $= \{V_1\}$

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



$k = 0$

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

$k = 1$

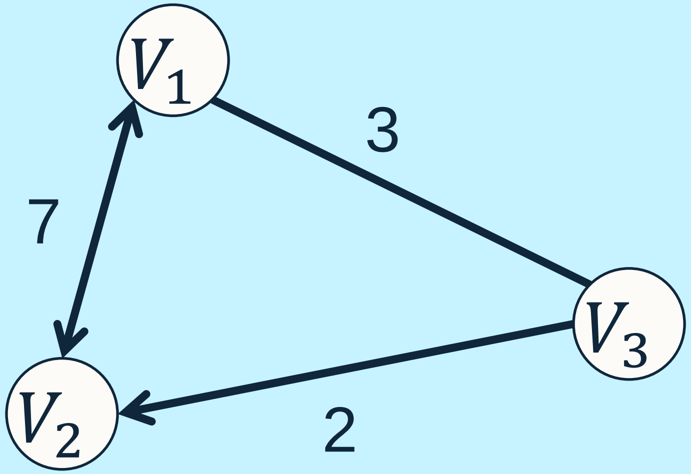
$i \backslash j$	V_1	V_2	V_3
V_1	0		
V_2			
V_3			

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 = $\{V_1\}$

$$\boxed{\begin{aligned} d(1,2,1) &= \min[d(1,2,0), d(1,1,0) + d(1,2,0)] \\ &= \min[7, 0+7] = 7 \end{aligned}}$$



$k = 0$

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

$k = 1$

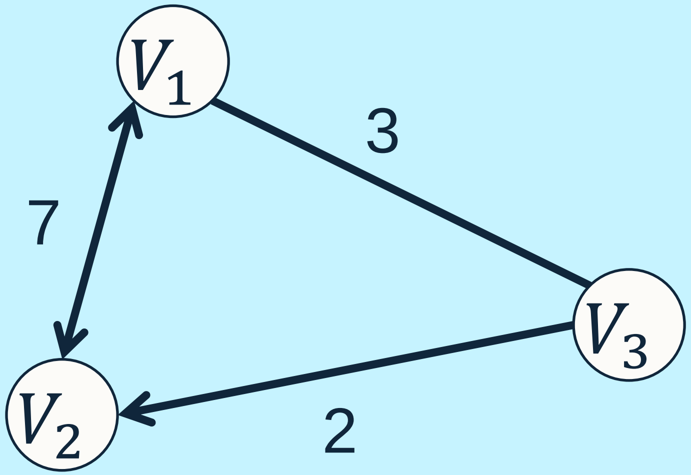
$i \backslash j$	V_1	V_2	V_3
V_1	0	7	
V_2			
V_3			

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

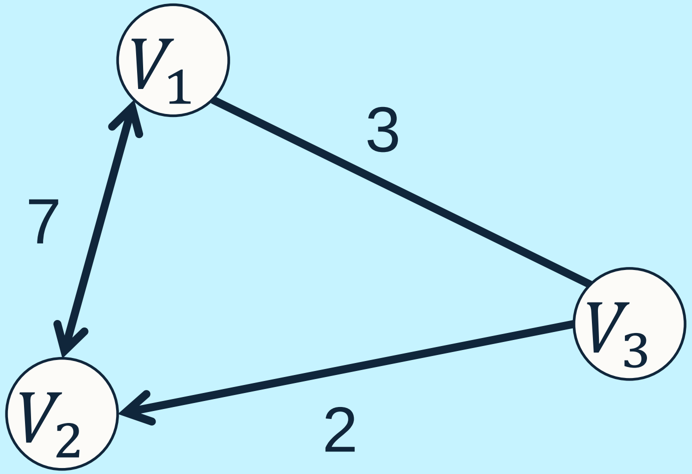
$k = 1$

$i \backslash j$	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

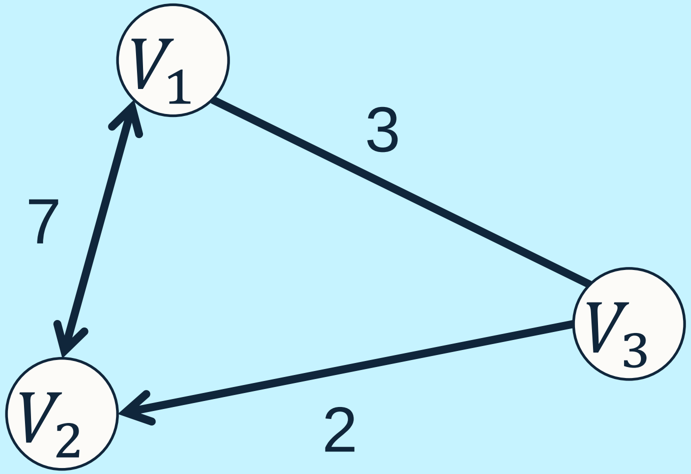
$k = 1$

$i \backslash j$	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…



$k = 0$

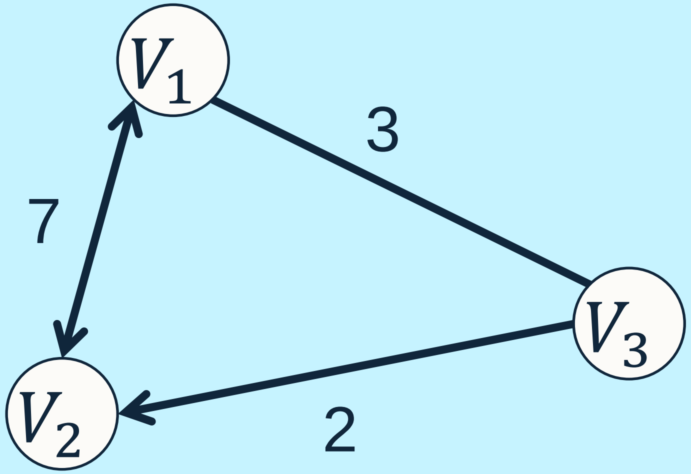
$i \backslash j$	V_1	V_2	V_3
V_1	0	7	3
V_2	7	0	∞
V_3	3	2	0

$k = 1$

$i \backslash j$	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 $= \{V_1\}$

$k = 0$

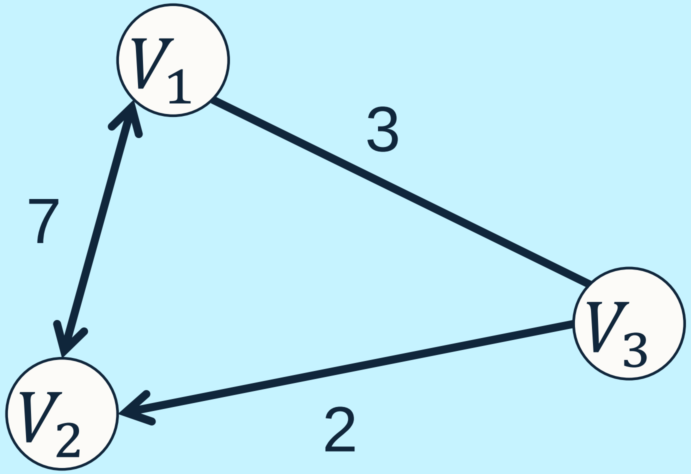
$i \backslash j$	V_1	V_2	V_3
V_1	0	7	3
V_2	7	0	∞
V_3	3	2	0

$k = 1$

$i \backslash j$	V_1	V_2	V_3
V_1			
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

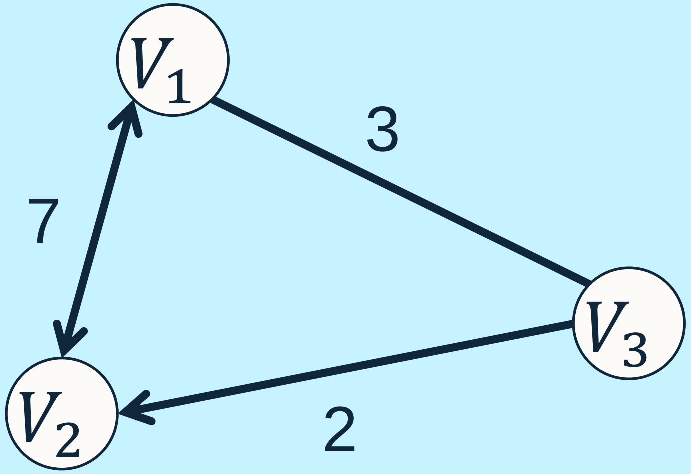
$i \backslash j$	V_1	V_2	V_3	
V_1				**0**
V_2				**7**
V_3				**3**
	0	**7**	**3**	

$k = 1$

$i \backslash j$	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

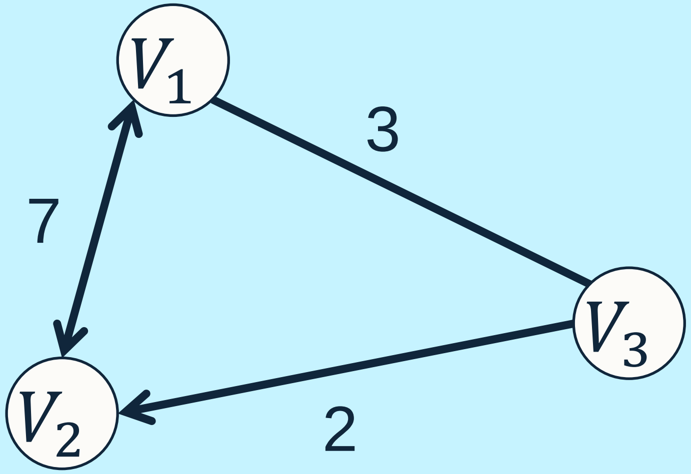
$i \backslash j$	V_1	V_2	V_3	
V_1	0+0	0+7	0+3	**0**
V_2	7+0	7+7	7+3	**7**
V_3	3+0	3+7	3+3	**3**
	0	**7**	**3**	

$k = 1$

$i \backslash j$	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

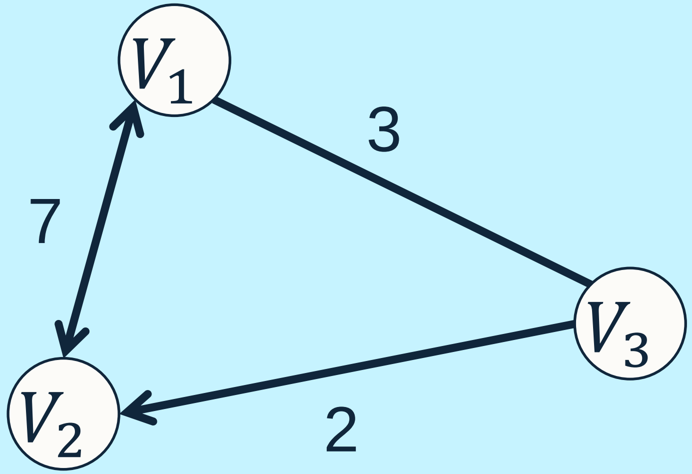
$i \backslash j$	V_1	V_2	V_3	
V_1	0	7	3	**0**
V_2	7	14	10	**7**
V_3	3	10	6	**3**
	0	**7**	**3**	

$k = 1$

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

Floyd-Warshall Example: Shortcut

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



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

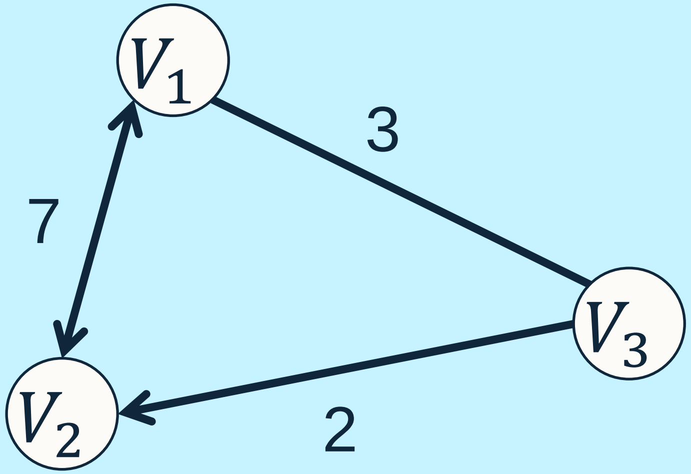
$i \backslash j$	V_1	V_2	V_3	
V_1	0	7	3	**0**
V_2	7	14	10	**7**
V_3	3	10	6	**3**
	0	**7**	**3**	

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

$i \backslash j$	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

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



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

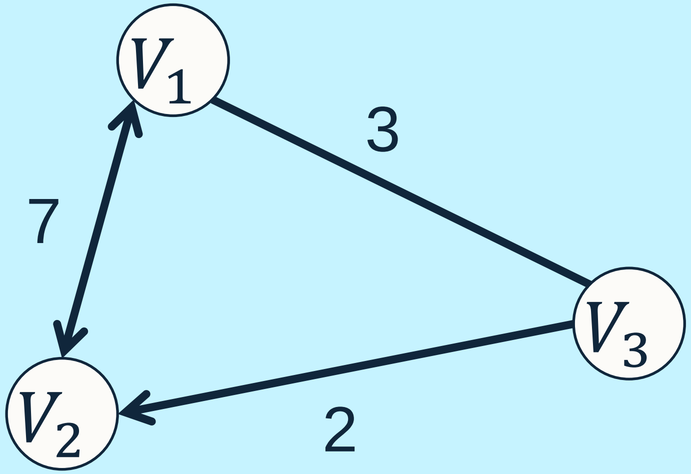
$i \backslash j$	V_1	V_2	V_3	
V_1				**7**
V_2				**0**
V_3				**2**
	7	**0**	**10**	

$k = 2$ **as** $\boldsymbol{min[k = 1}$, **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

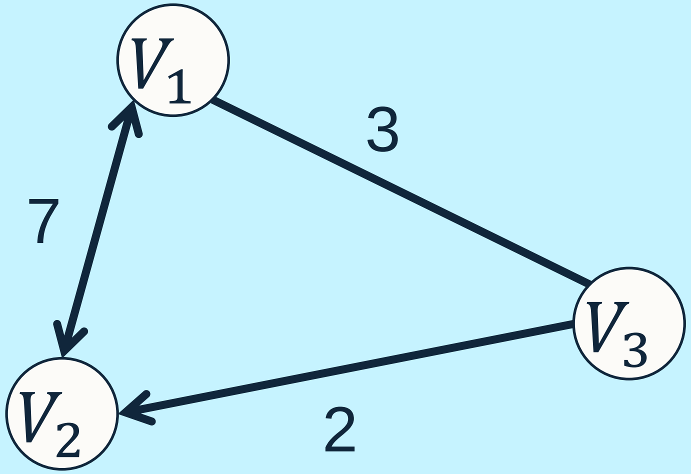
$i \backslash j$	V_1	V_2	V_3	
V_1	14	7	17	**7**
V_2	7	0	10	**0**
V_3	9	2	12	**2**
	7	**0**	**10**	

$k = 2$ **as** $\boldsymbol{min[k = 1,}$ **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

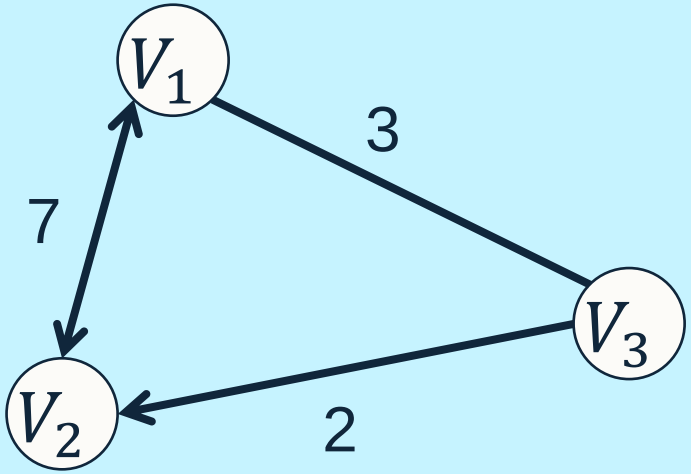
$i \backslash j$	V_1	V_2	V_3	
V_1	14	7	17	**7**
V_2	7	0	10	**0**
V_3	9	2	12	**2**
	7	**0**	**10**	

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

$i \backslash j$	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

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



- $k = 3$
- $d(i,j,3) = \min[d(i,j,2), \boldsymbol{d(i,3,2)} + \boldsymbol{d(3,j,2)}]$
- 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** $\boldsymbol{min[k = 2, \text{Intermediate sum}]}$

$i \backslash j$	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
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

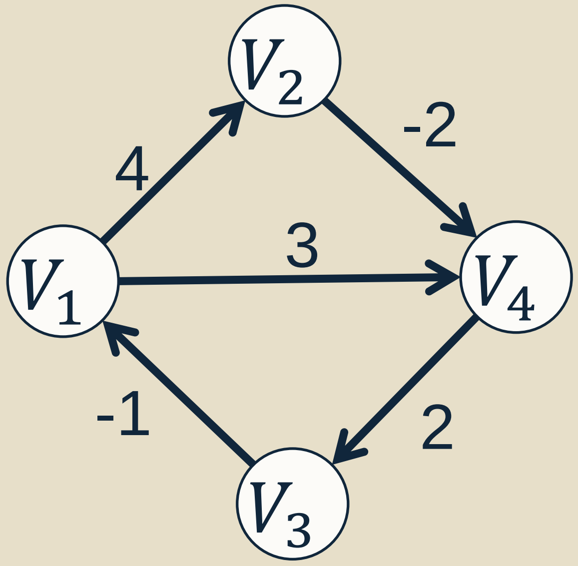
$k = 3$

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

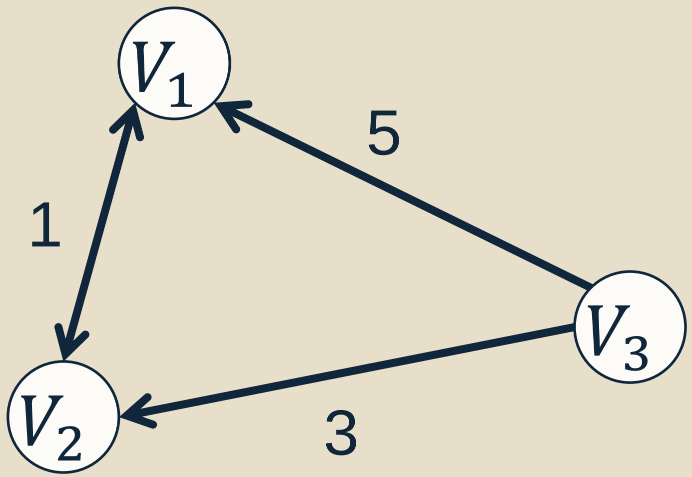
Floyd-Warshall Questions

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

Q1.



Q2.



University of Nottingham
UK | CHINA | MALAYSIA

Thank you