Faculty of Computer Science
Data Science and Business Analytics (DSBA)

Algorithms and Data Structures

Seminar 10 – Module 3

Floyd-Warshall Algorithm
"All-Pairs Optimal Cost Paths"

February 2022

J.C. Carrasquel

Floyd-Warshall Algorithm
"All-Pairs Minimum Cost Paths"

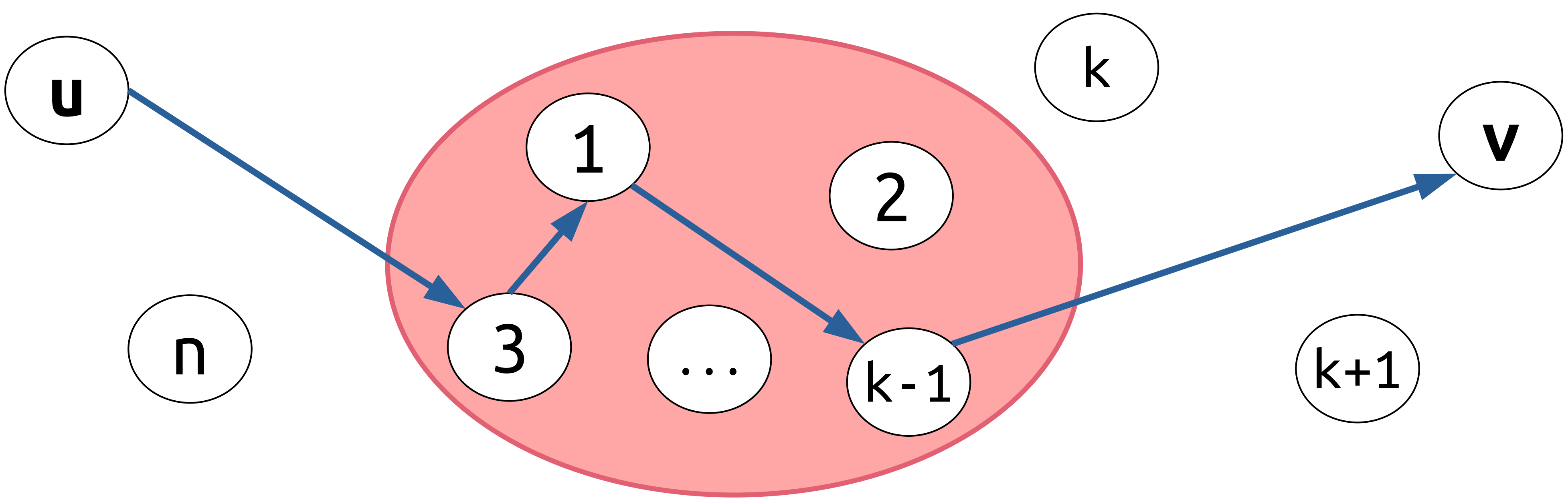
Floyd-Warshall's Algorithm follows a ***dynamic programming*** approach

Let us label nodes as 1, 2,…, n

Sub-Problem of size k

For all possible pairs of nodes u,v in the graph, calculate $\mathbf{D^{(k-1)}[u][v]}$

$\mathbf{D^{(k-1)}[u][v]}$: cost of the optimal path from node u to node v, so that for every intermediate node x in the path we have that $x \in \{1, \ldots, k-1\}$

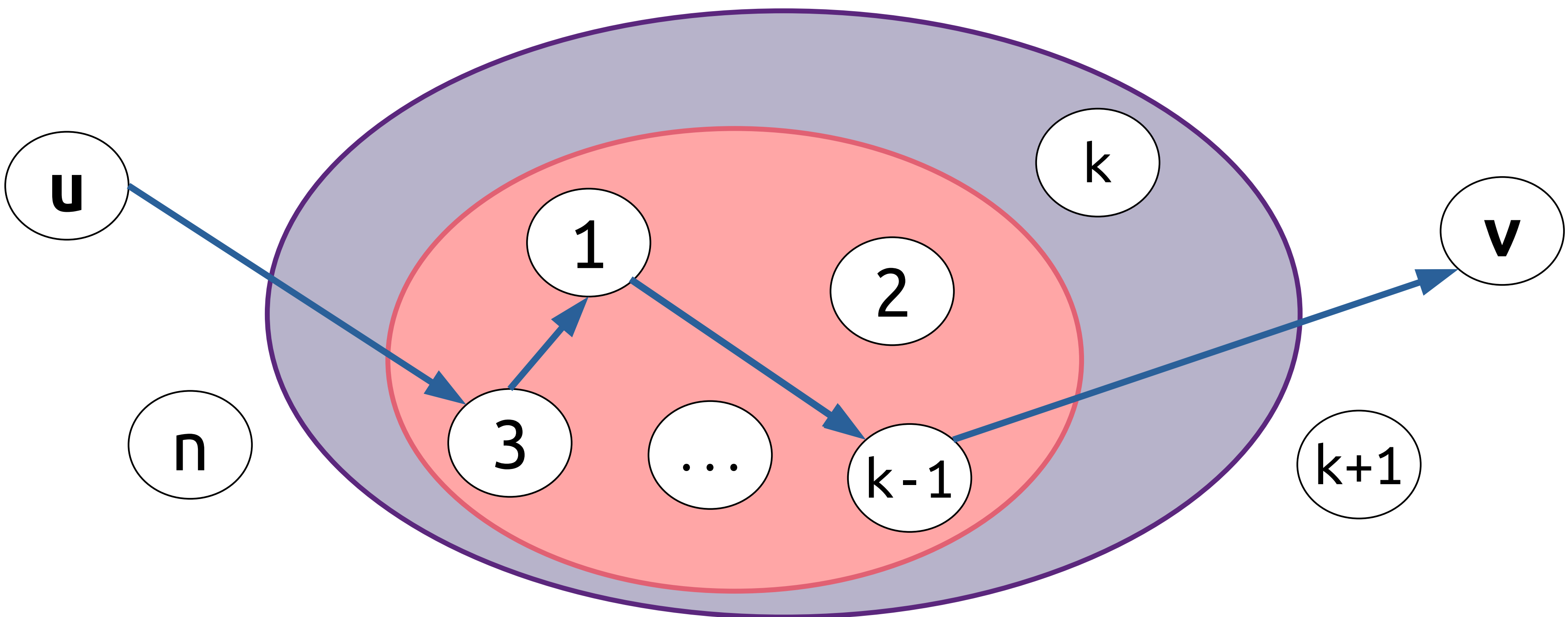


Floyd-Warshall Algorithm
"All-Pairs Minimum Cost Paths"

Floyd-Warshall's Algorithm follows a ***dynamic programming*** approach

How we can find $D^{(k)}[u][v]$ using $D^{(k-1)}[u][v]$?

$D^{(k)}[u][v]$: cost of the optimal path from node u to node v, so that for every intermediate node x in the path we have that $x \in \{1, \dots, k\}$



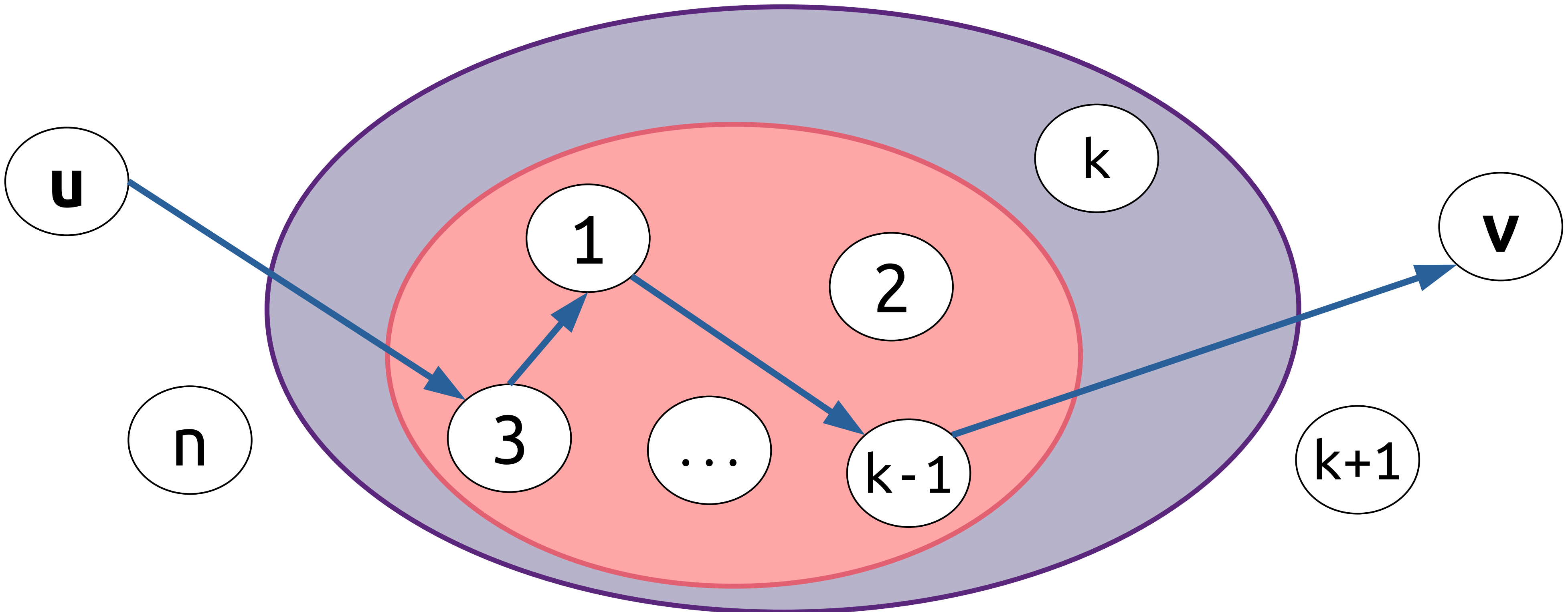
Floyd-Warshall Algorithm
“All-Pairs Minimum Cost Paths”

Floyd-Warshall’s Algorithm follows a ***dynamic programming*** approach

How we can find $D^{(k)}[u][v]$ using $D^{(k-1)}[u][v]$?

$D^{(k)}[u][v]$: cost of the optimal path from node u to node v, so that for every intermediate node x in the path we have that $x \in \{1, \dots, k\}$

Case #1: we DON’T need node k. $D^{(k)}[u][v] = D^{(k-1)}[u][v]$



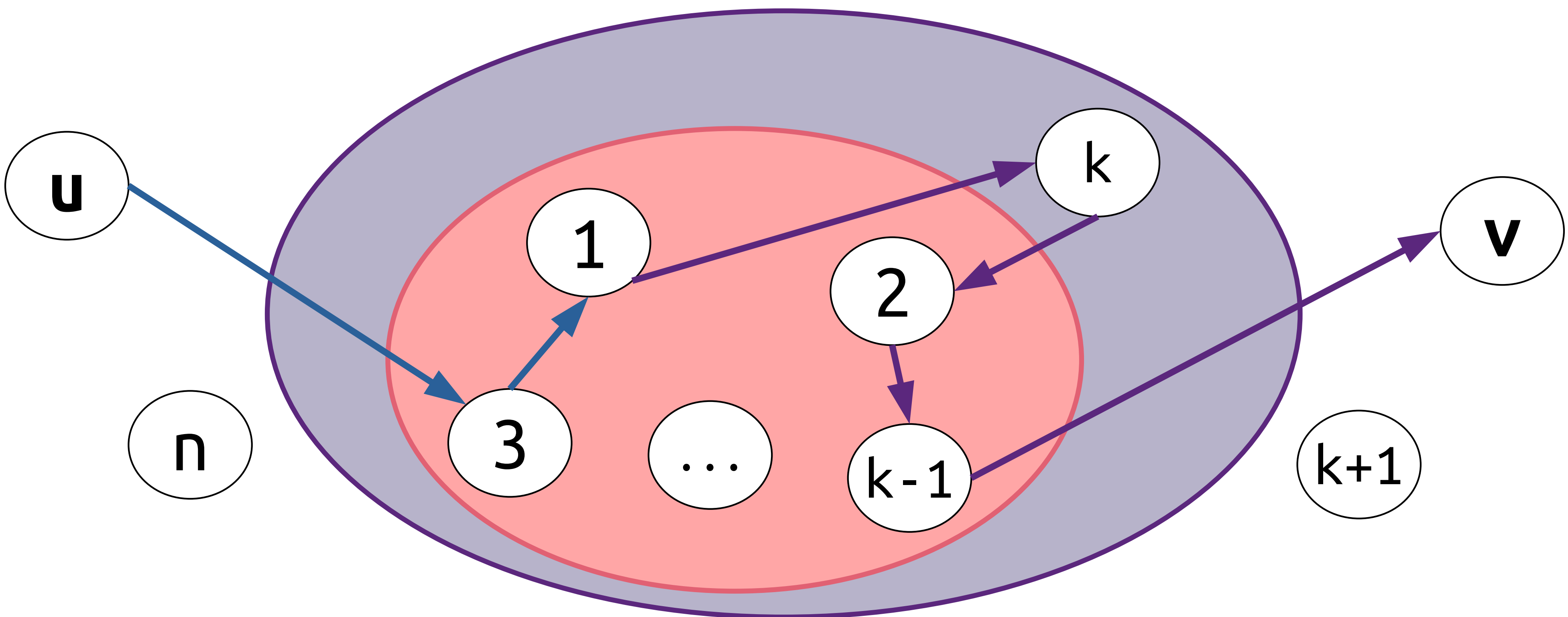
Floyd-Warshall Algorithm
"All-Pairs Minimum Cost Paths"

Floyd-Warshall's Algorithm follows a ***dynamic programming*** approach

How we can find $D^{(k)}[u][v]$ using $D^{(k-1)}[u][v]$?

$D^{(k)}[u][v]$: cost of the optimal path from node u to node v, so that for every intermediate node x in the path we have that $x \in \{1, \dots, k\}$

Case #2: we DO need node k. $D^{(k)}[u][v] = \mathbf{D^{(k-1)}[u][k]} + \mathbf{D^{(k-1)}[k][v]}$



Floyd-Warshall Algorithm
"All-Pairs Minimum Cost Paths"

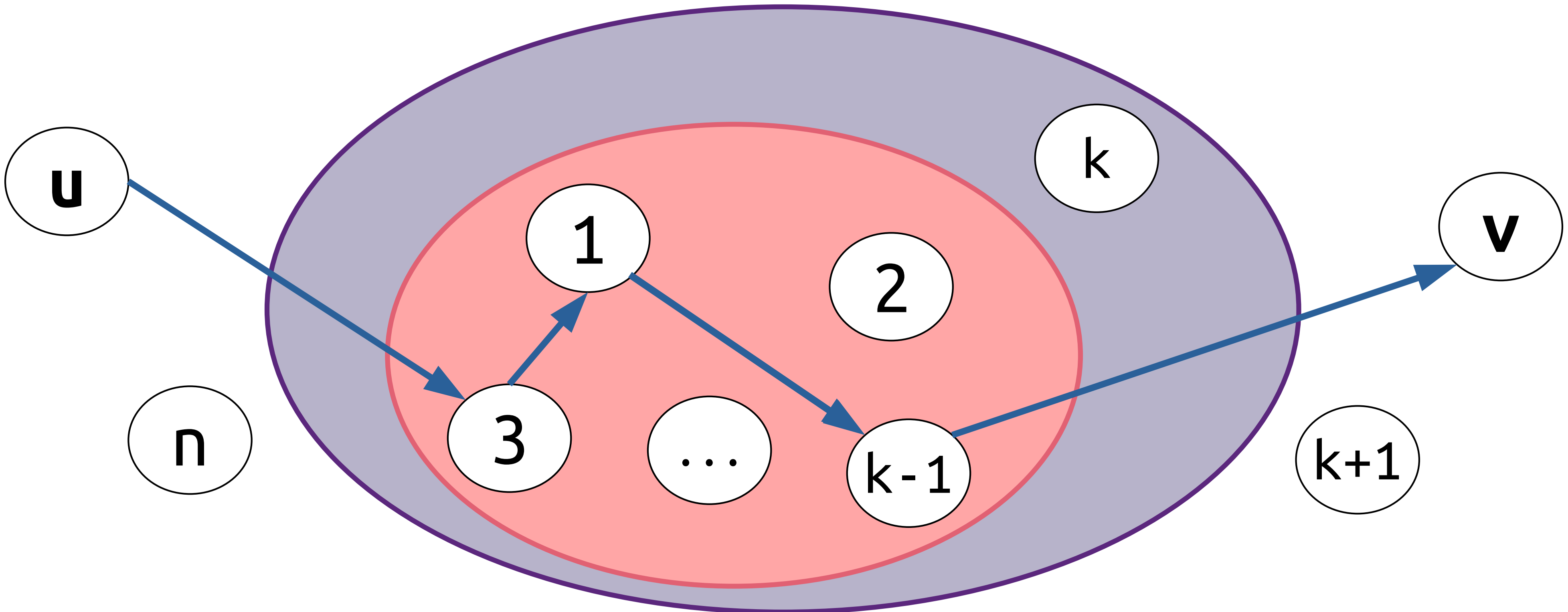
Floyd-Warshall's Algorithm follows a ***dynamic programming*** approach

How we can find $D^{(k)}[u][v]$ using $D^{(k-1)}[u][v]$?

$D^{(k)}[u][v]$: cost of the optimal path from node u to node v, so that for every intermediate node x in the path we have that $x \in \{1, \dots, k\}$

Putting Case#1 and Case#2 together!

$$\boxed{D^{(k)}[u][v] = \min(\, D^{(k-1)}[u][v] \,,\; D^{(k-1)}[u][k] \;+\; D^{(k-1)}[k][v] \,)}$$



Floyd-Warshall Algorithm
"All-Pairs Minimum Cost Paths"

Floyd-Warshall's Algorithm follows a ***dynamic programming*** approach

Idea: For a graph $G = (V,E)$, let $n = |V|$

Initialize n-by-n matrices $D^{(k)}$ for $k = 0,\ldots,n$
- $D^{(k)}[u][u] = 0$ for all u, for all k
- $D^{(k)}[u][v] = \infty$ for all $u \neq v$, for all k
- $D^{(0)}[u][v] = \text{edgeWeight}(u,v)$ for all $(u,v) \in E$

For k=1,…,n:

For each pair (u,v) in VxV:

$$D^{(k)}[u][v] = \min(D^{(k-1)}[u][v] ,\ D^{(k-1)}[u][k] + D^{(k-1)}[k][v])$$

return $D^{(n)}$