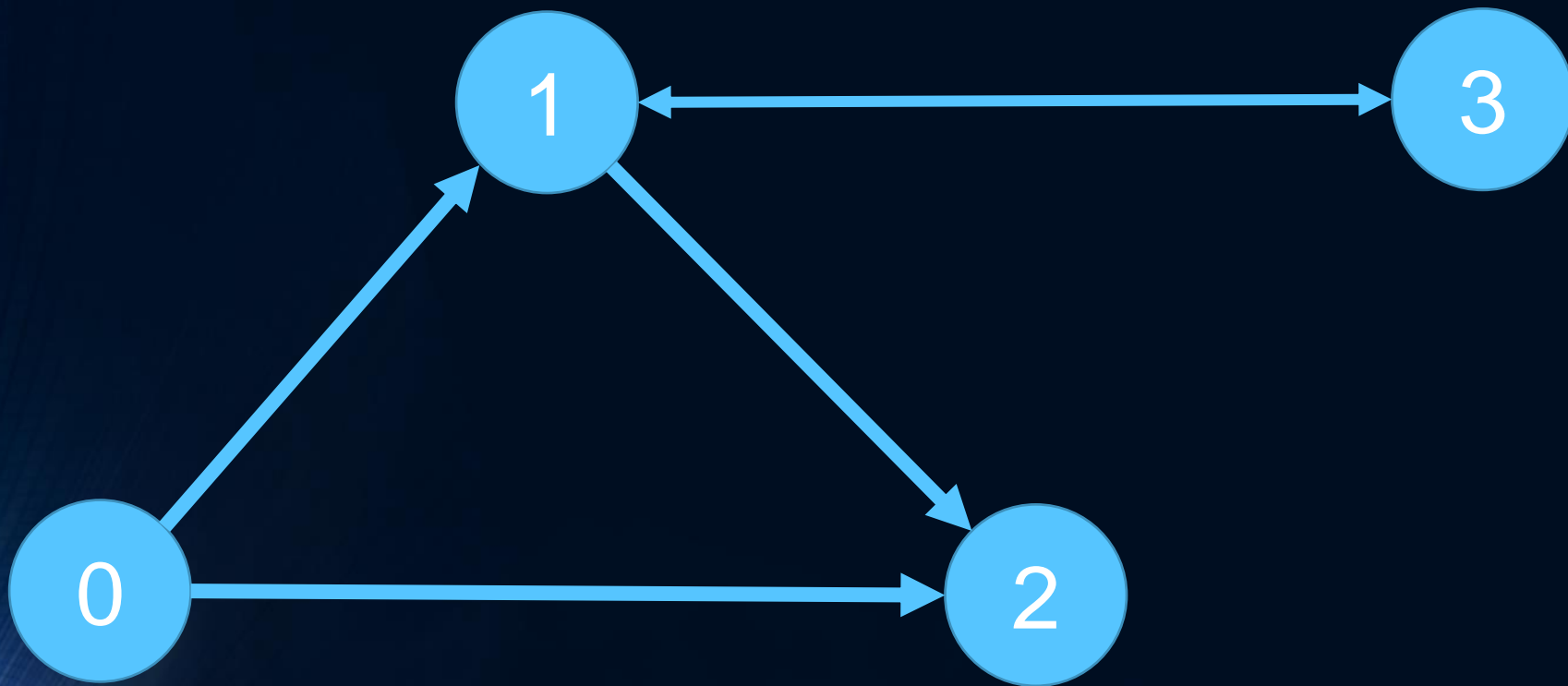


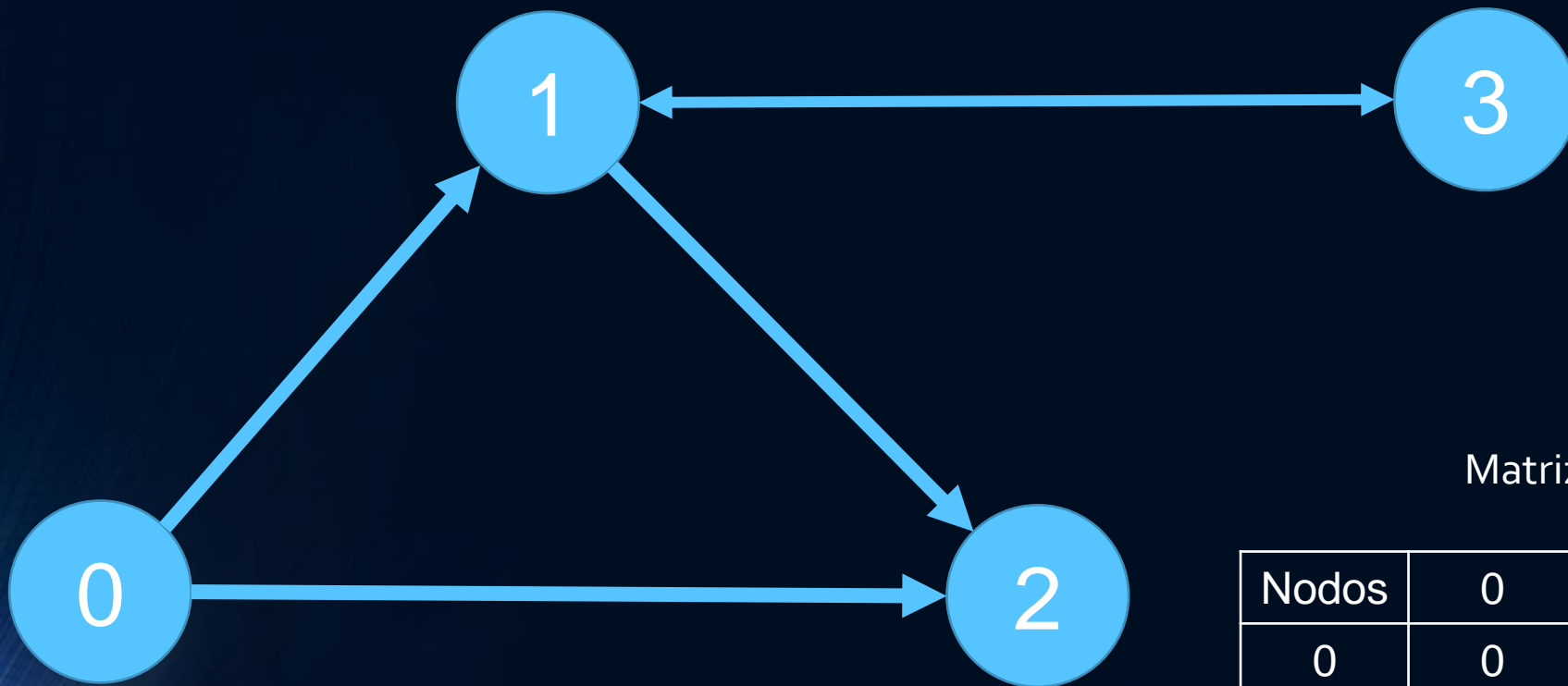
# Grafos

ANDRÉS SÁNCHEZ CABRERA  
AND.SANCHEZ@UCSC.CL

# Definición

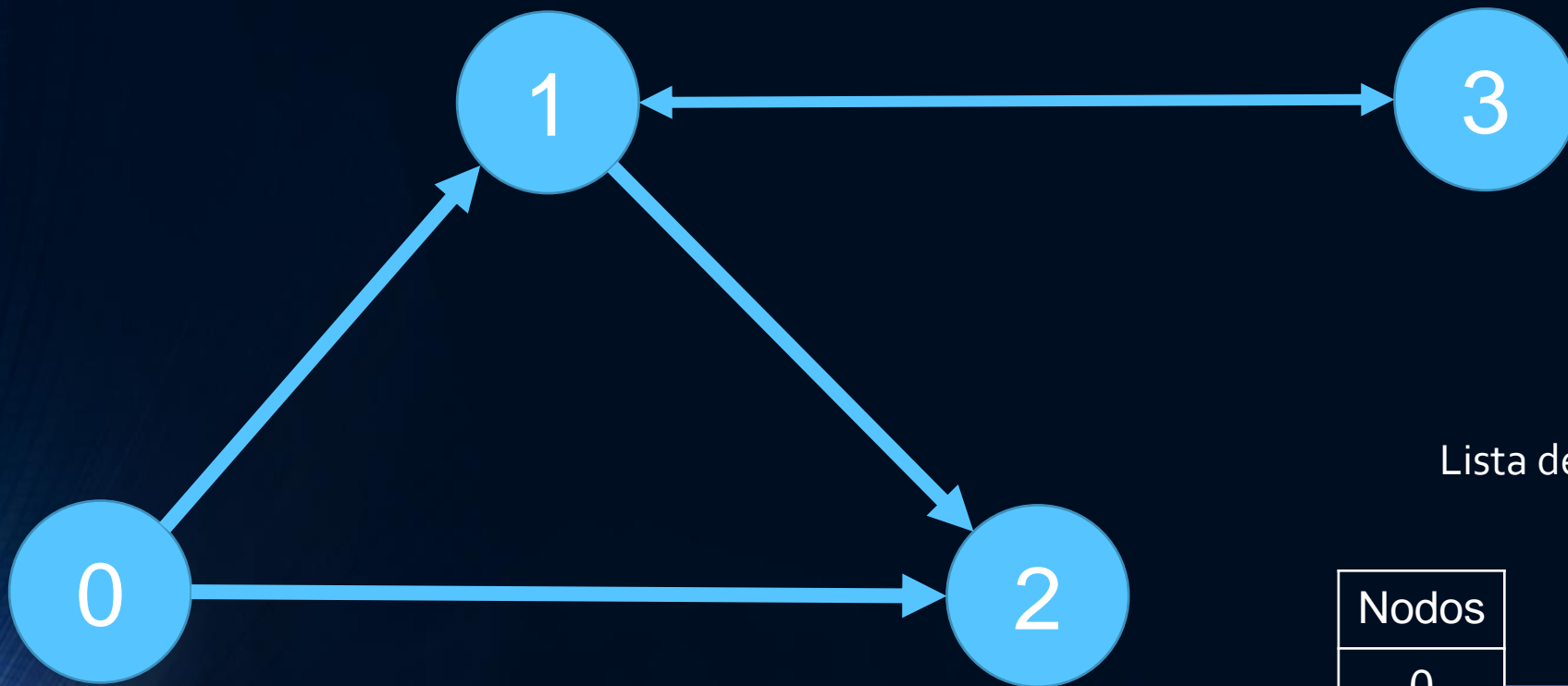
- Un grafo es una estructura de datos que esta compuesta por un conjunto de vértices (usualmente llamados nodos), y un conjunto de aristas (arcos o caminos), en donde las aristas representan una relación entre 2 nodos, y los nodos poseen un conjunto de información.





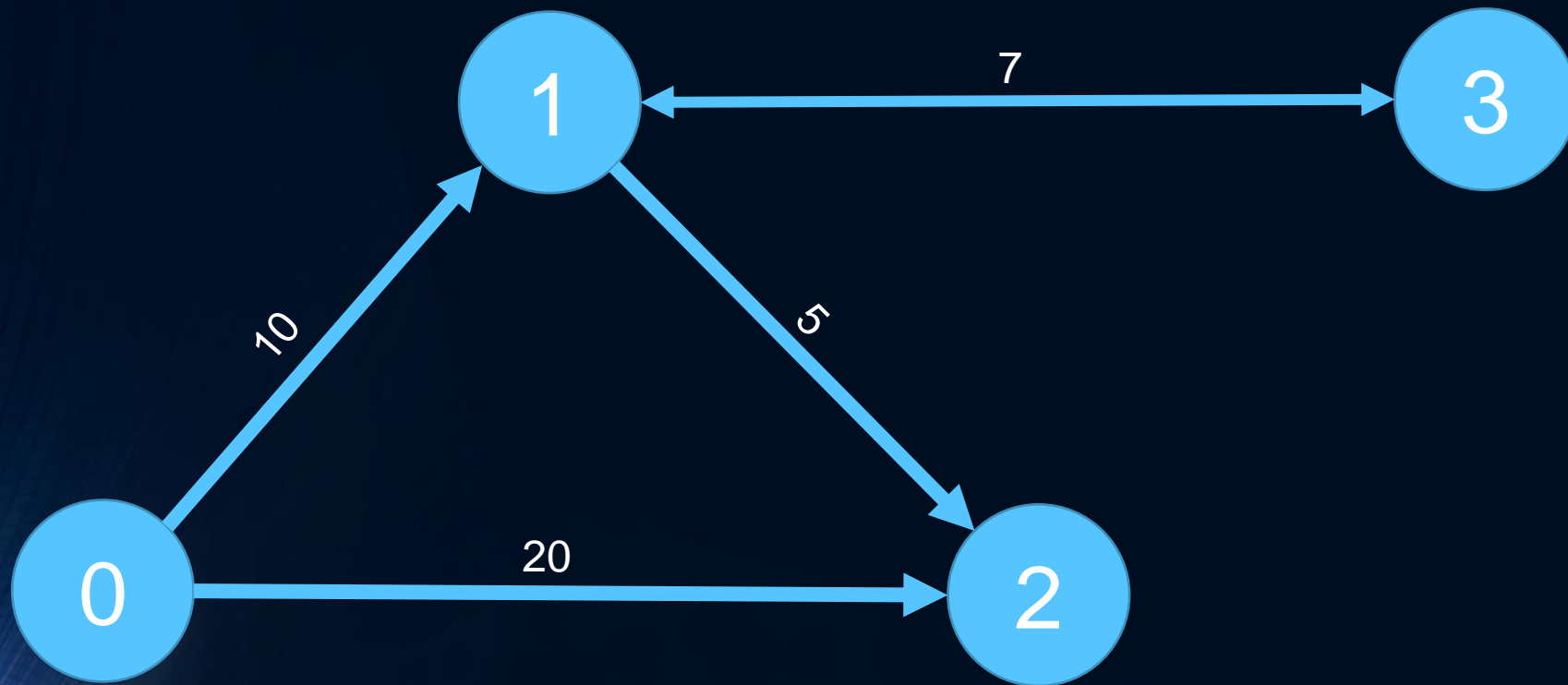
Matriz de adyacencia

Nodos	0	1	2	3
0	0	1	1	0
1	0	0	1	1
2	0	0	0	0
3	0	1	0	0

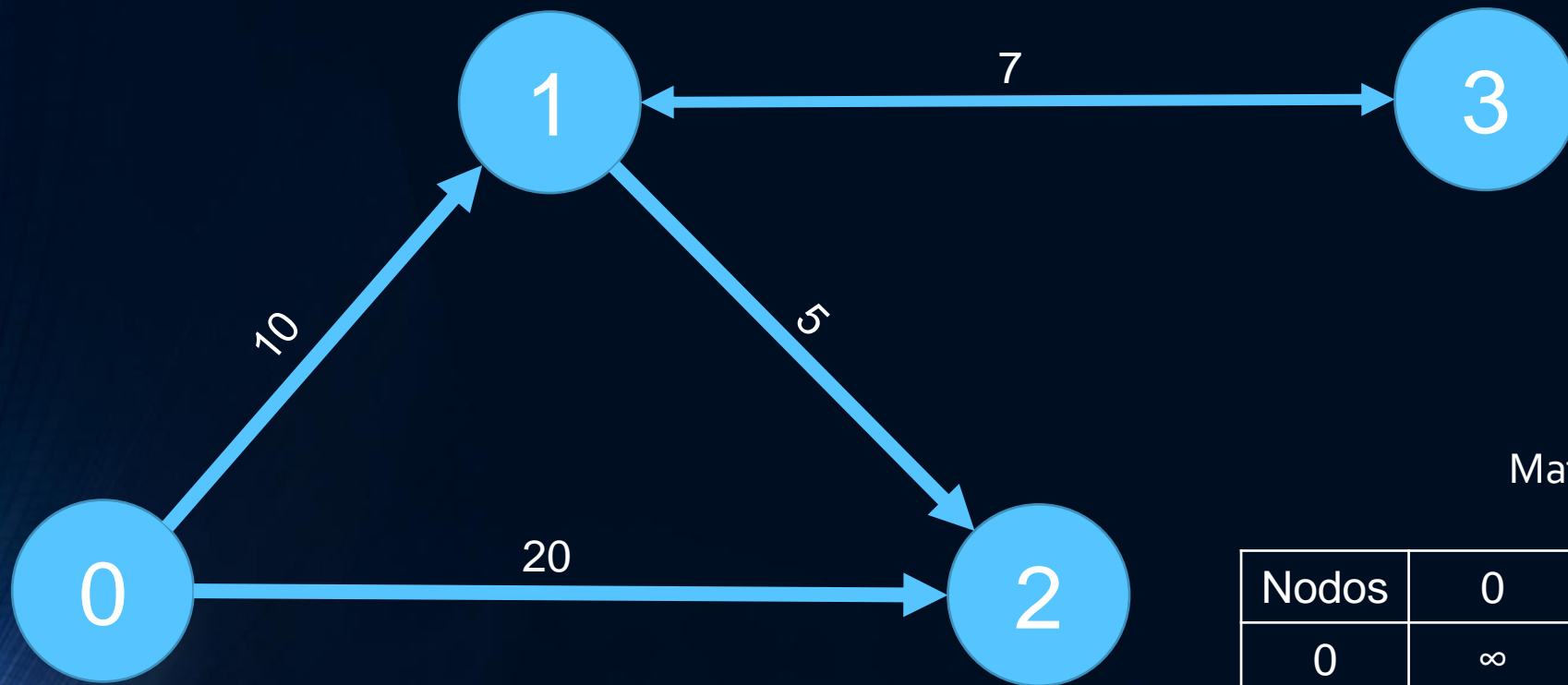


Lista de adyacencia

Nodos		
0	→	1 2
1	→	2 3
2		
3	→	1

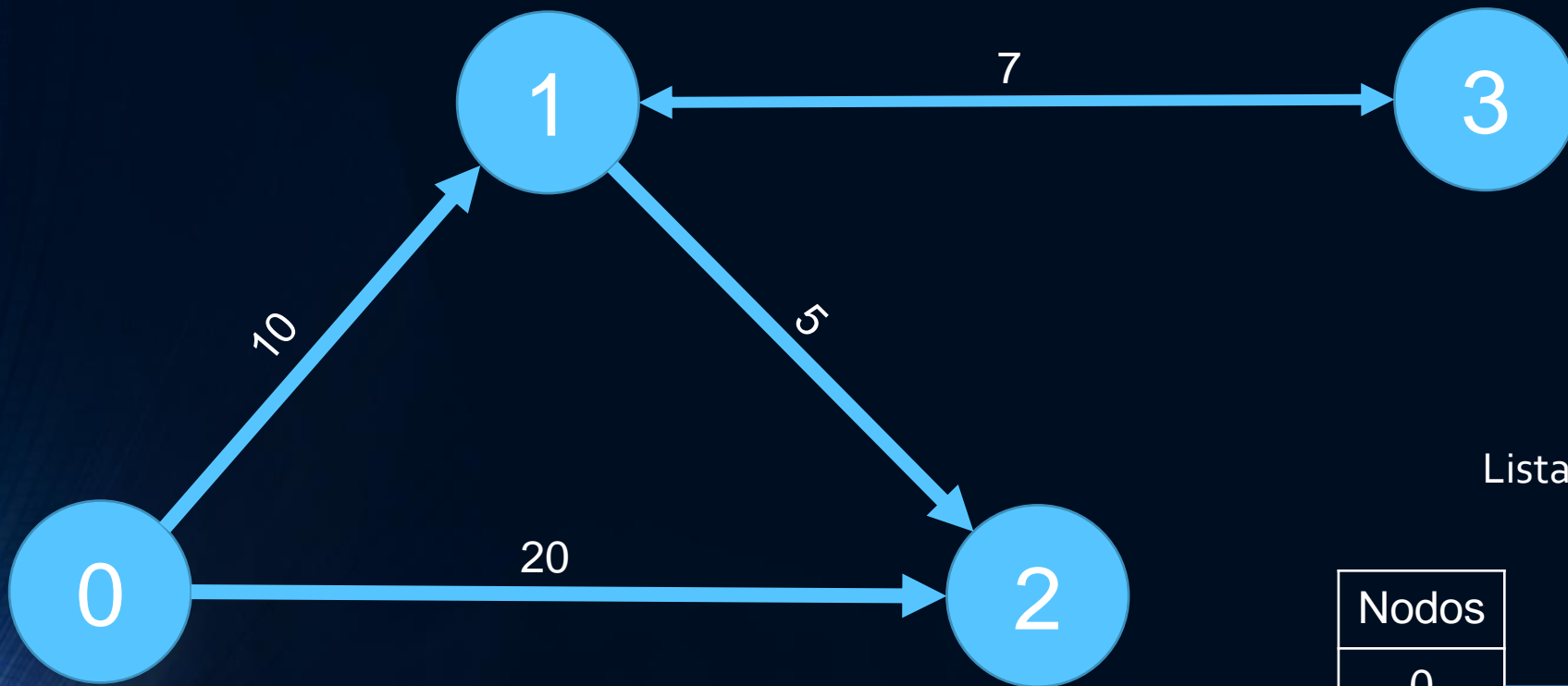






Matriz de costos

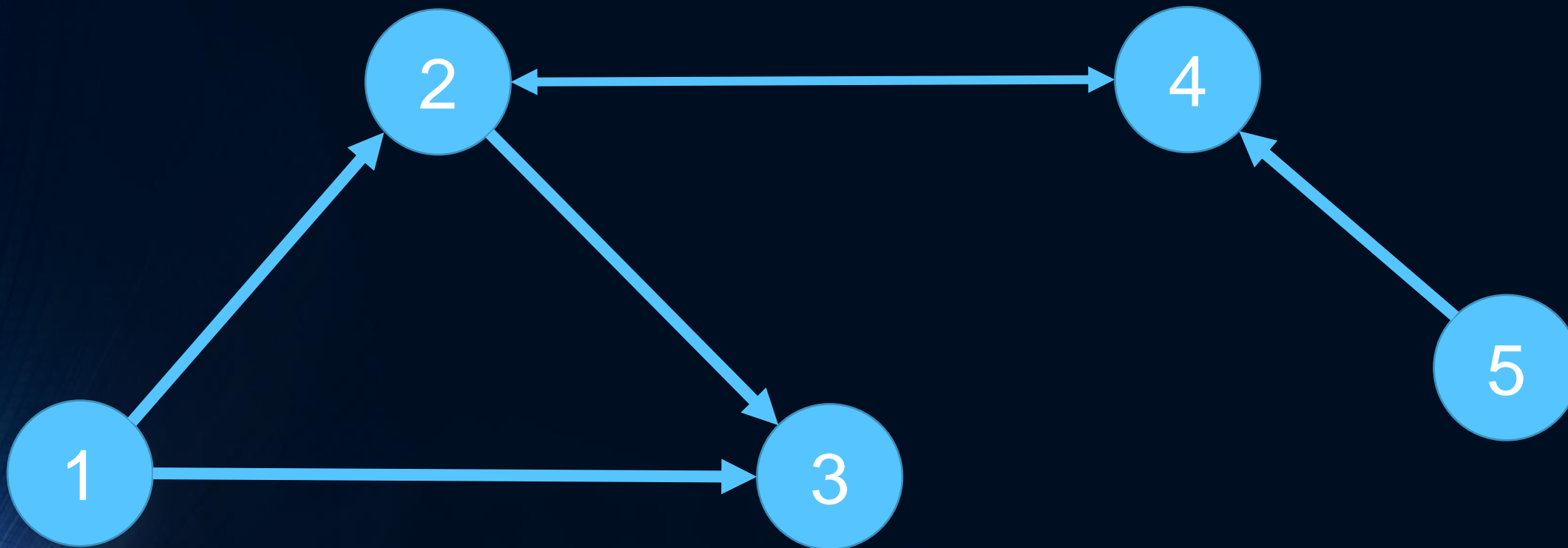
Nodos	0	1	2	3
0	$\infty$	10	20	$\infty$
1	$\infty$	$\infty$	5	7
2	$\infty$	$\infty$	$\infty$	$\infty$
3	$\infty$	7	$\infty$	$\infty$



Lista de costos

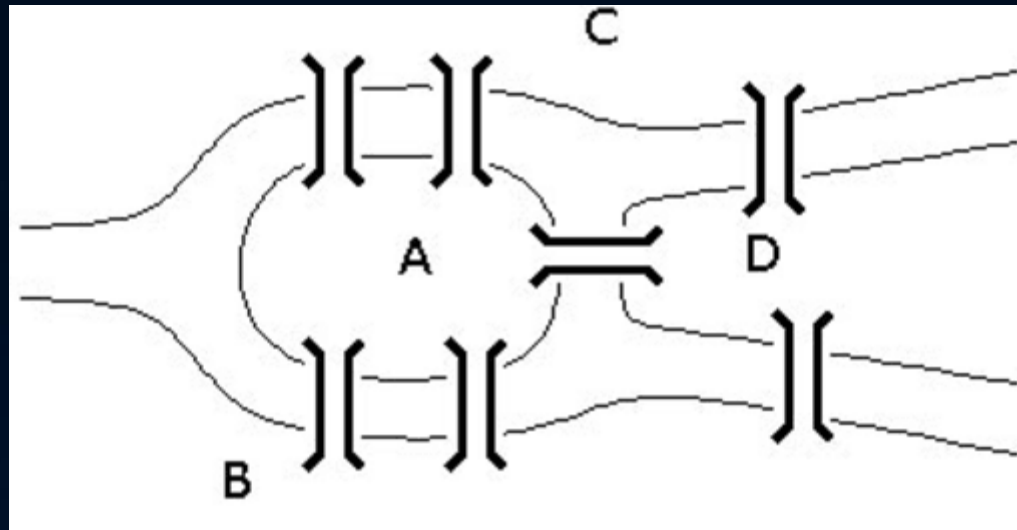
Nodos		
0	→	(1, 10) (2, 20)
1	→	(2, 5) (3, 7)
2		
3	→	(1, 7)



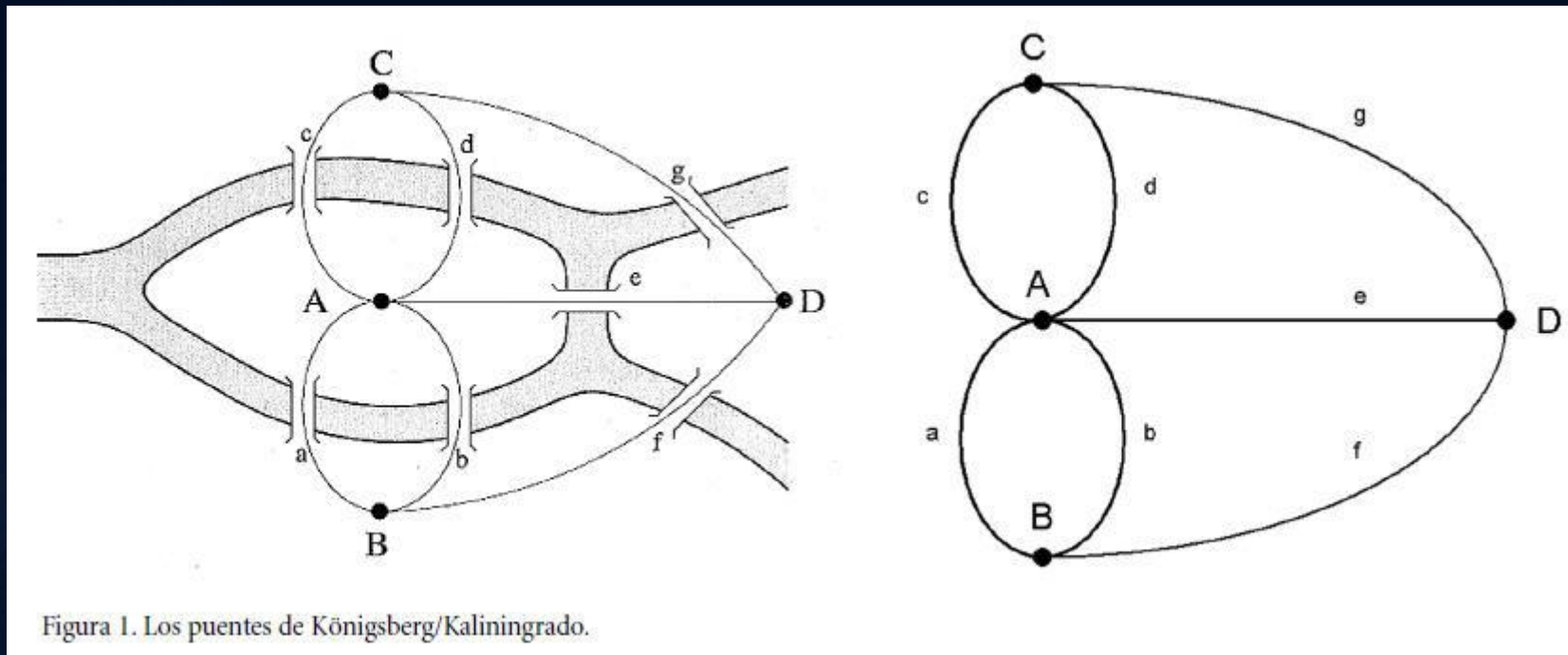


# ¿Cuál es la dificultad?

- La dificultad radica en que no siempre nos darán el grado de una forma simple, muchas veces deberemos abstraernos y encontrar la forma de crear nuestro grafo, un ejemplo de esto, son los puentes de Königsberg.



# Puentes de Königsberg



## Ruta de números

0	3	1	2	9
7	3	4	9	9
1	7	5	5	3
2	3	4	2	5

## Numerar los nodos

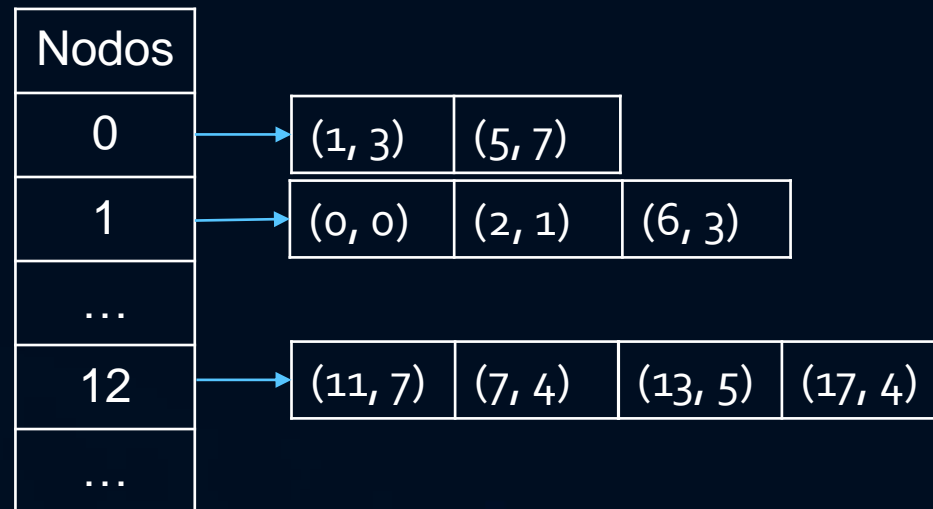
0	1	2	3	4
5	6	7	8	9
10	11	12	13	14
15	16	17	18	19

## Ruta de números

0	3	1	2	9
7	3	4	9	9
1	7	5	5	3
2	3	4	2	5



# Construyendo el grafo



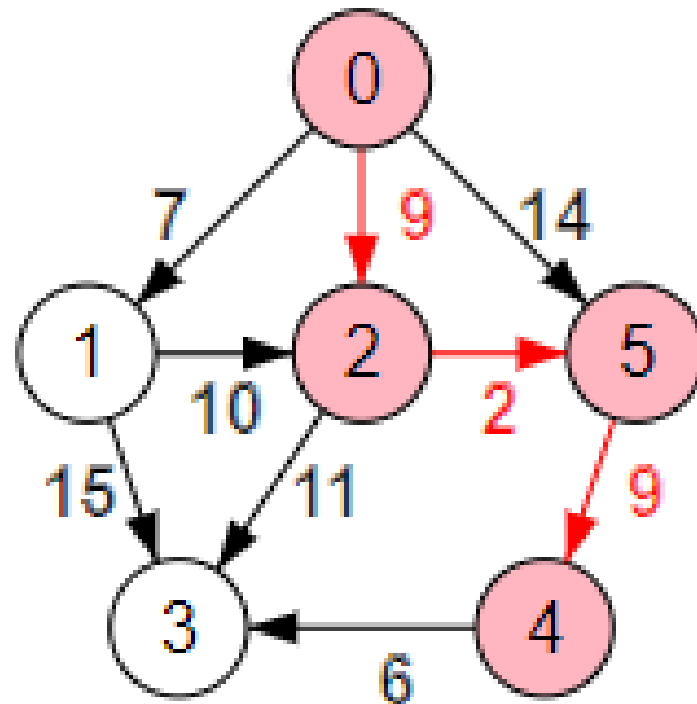
# Algunos algoritmos importantes

- BFS (Recorre grafos en anchura)
- DFS (Recorre grafos en profundidad)
- Dijkstra (Camino mínimo desde un nodo de inicio)
- Kruskal (Árbol de cobertura mínimo utilizando Union find)
- Prim (Árbol de cobertura mínimo)
- Bellman Ford's (Detecta ciclos negativo)

# Algunos algoritmos importantes

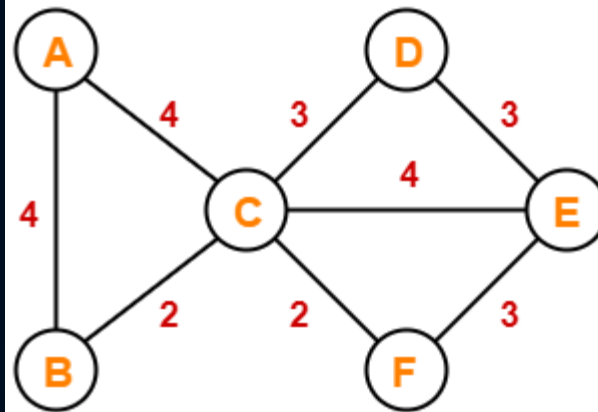
- Floyd Warshall (Distancia de todos los nodos a todos los nodos)
- Tarjan's Strongly connected components
- Dinic (Flujo máximo de un de un grafo desde una fuente a un destino)
- Bipartite Check (Detecta si es un grafo bipartito)
- Flood Fill (Colorea un grafo)
- Topological Sort (Ordena los nodos de un grafo de forma topológica)

# Dijkstra

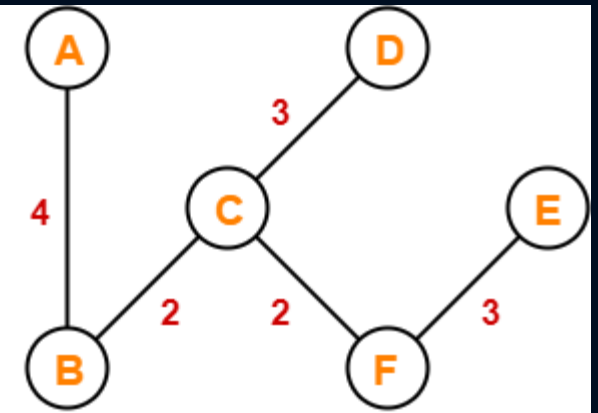


# Prim - Kruskal

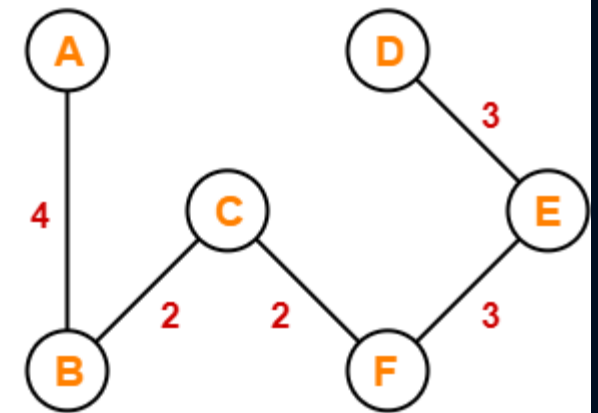
- Ambos algoritmos sirven para lo mismo.



Given Graph

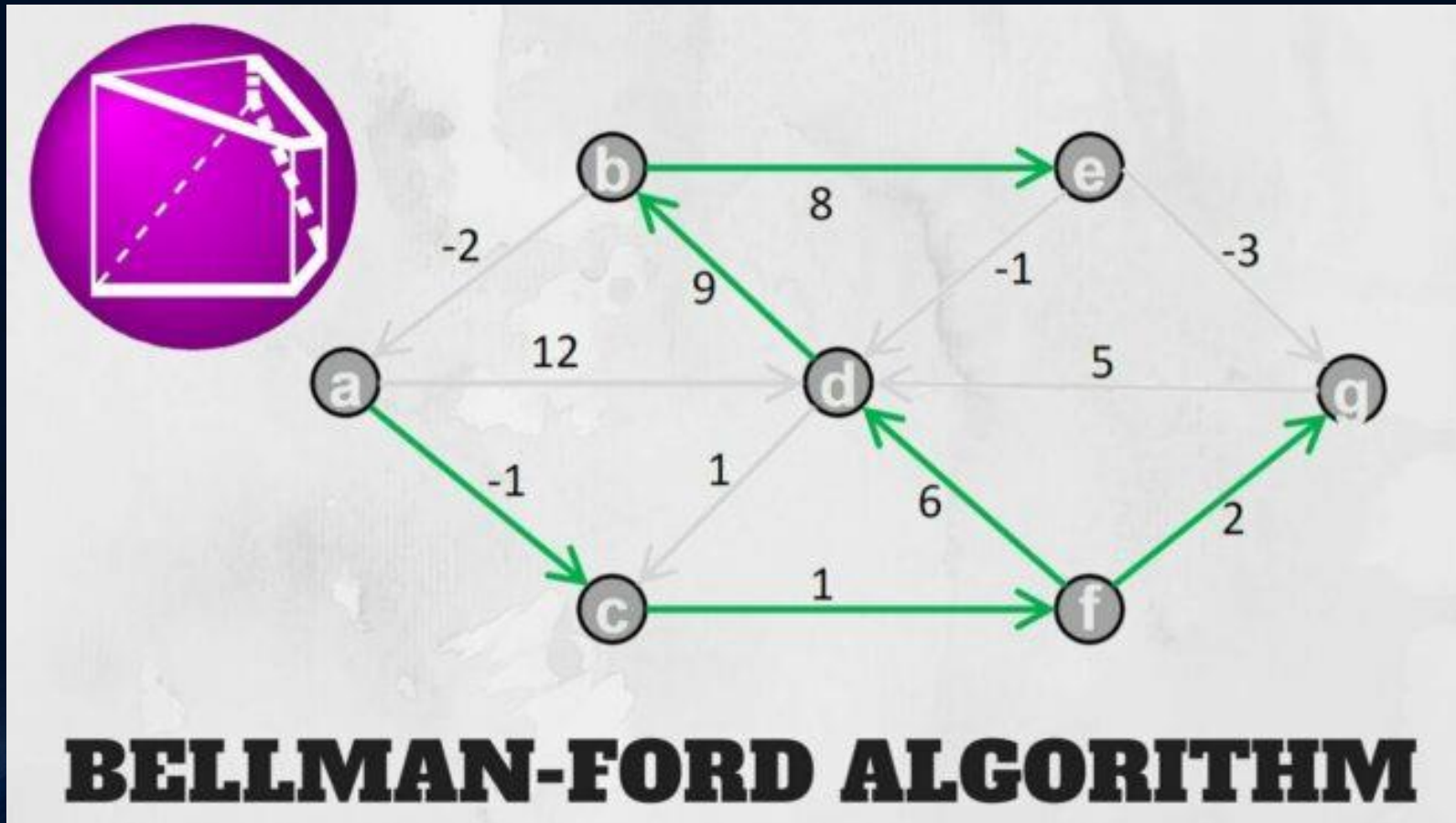


Result from Prim's Algorithm  
( Cost = 14 units )



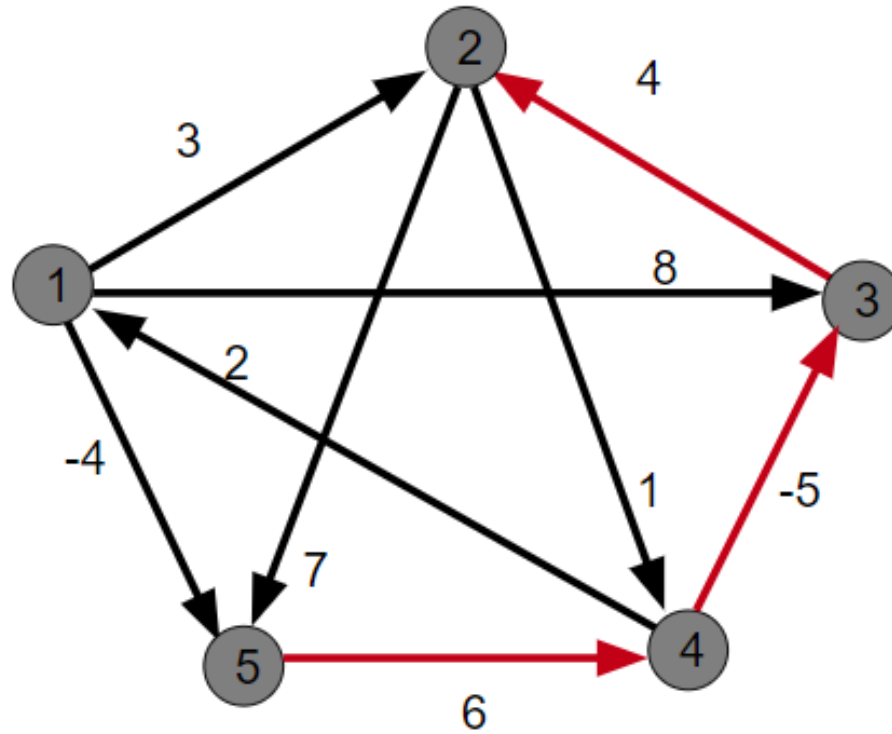
Result from Kruskal's Algorithm  
( Cost = 14 units )

# Bellman Ford - Cobertura





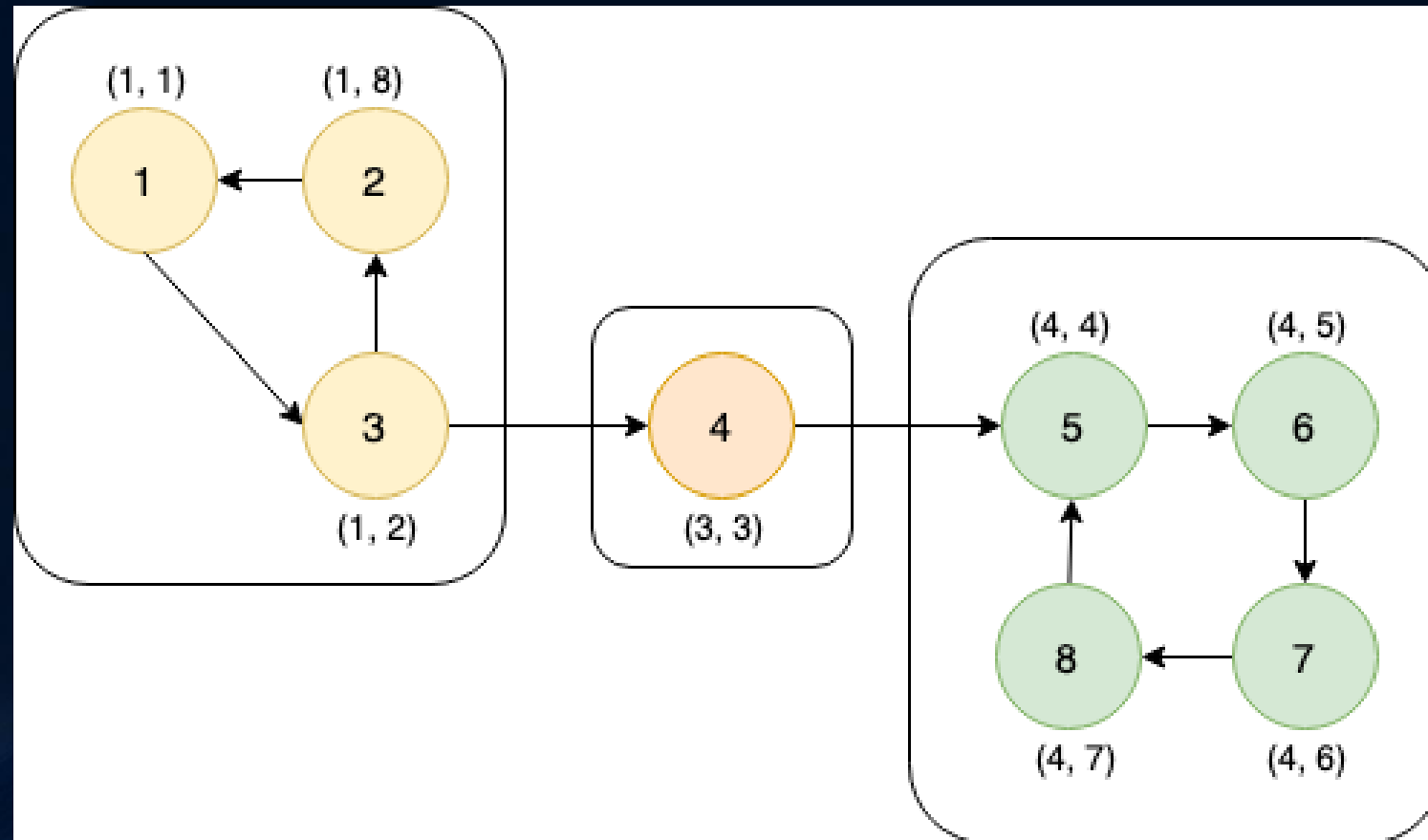
# Floyd Warshall



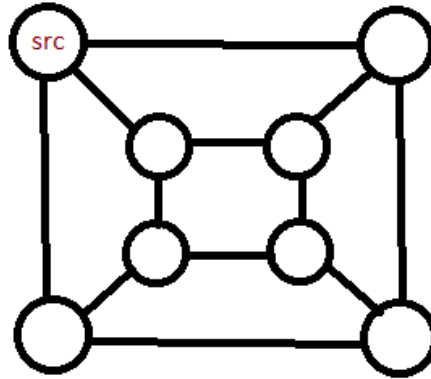
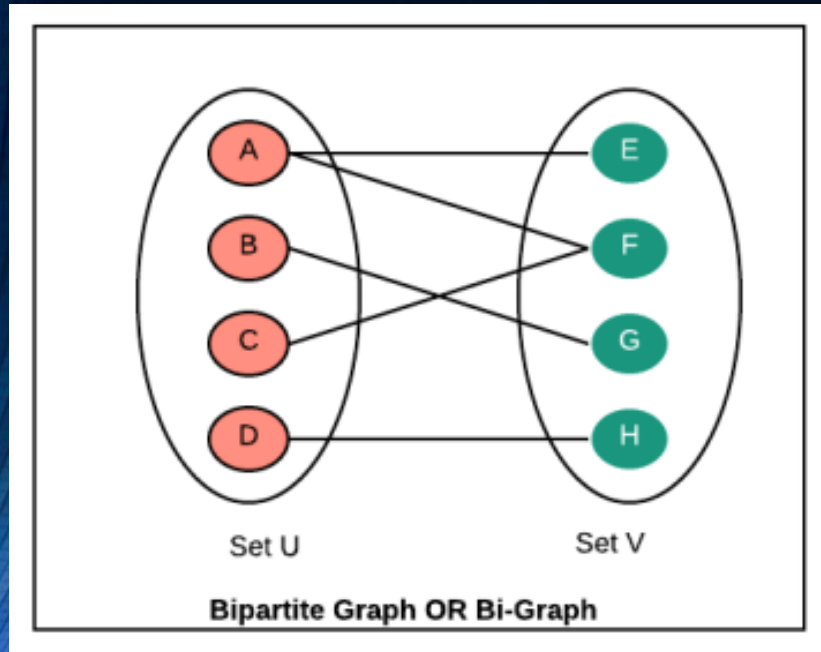
$D$

	1	2	3	4	5
1	0	1	-3	2	-4
2	3	0	-4	1	-1
3	7	4	0	5	3
4	2	-1	-5	0	-2
5	8	5	1	6	0

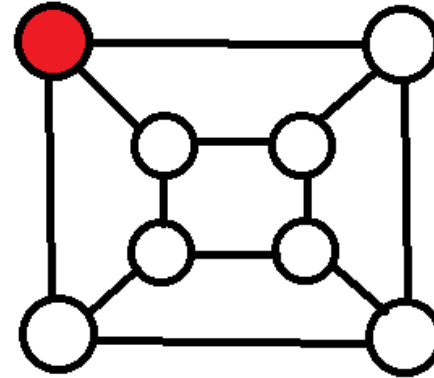
# Tarjan's Strongly connected components



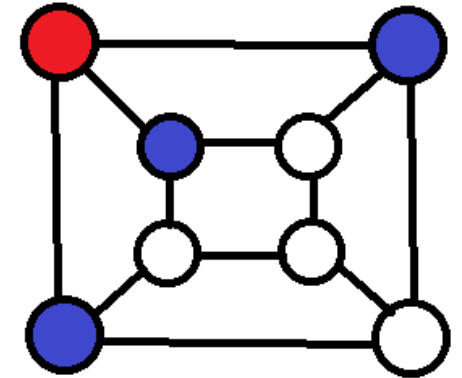
# Bipartite Check



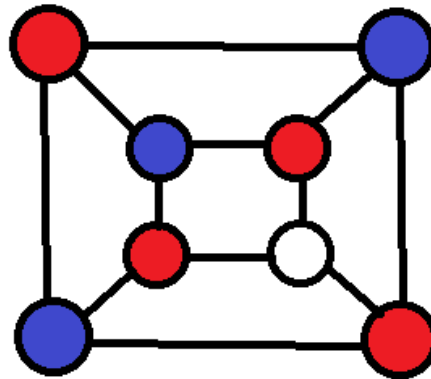
given a graph with source vertex



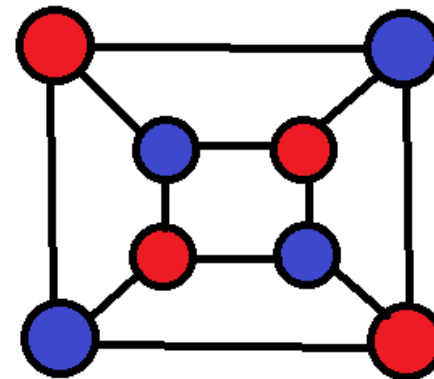
colour src vertex, say red



assign another colour to the neighbours, say blue



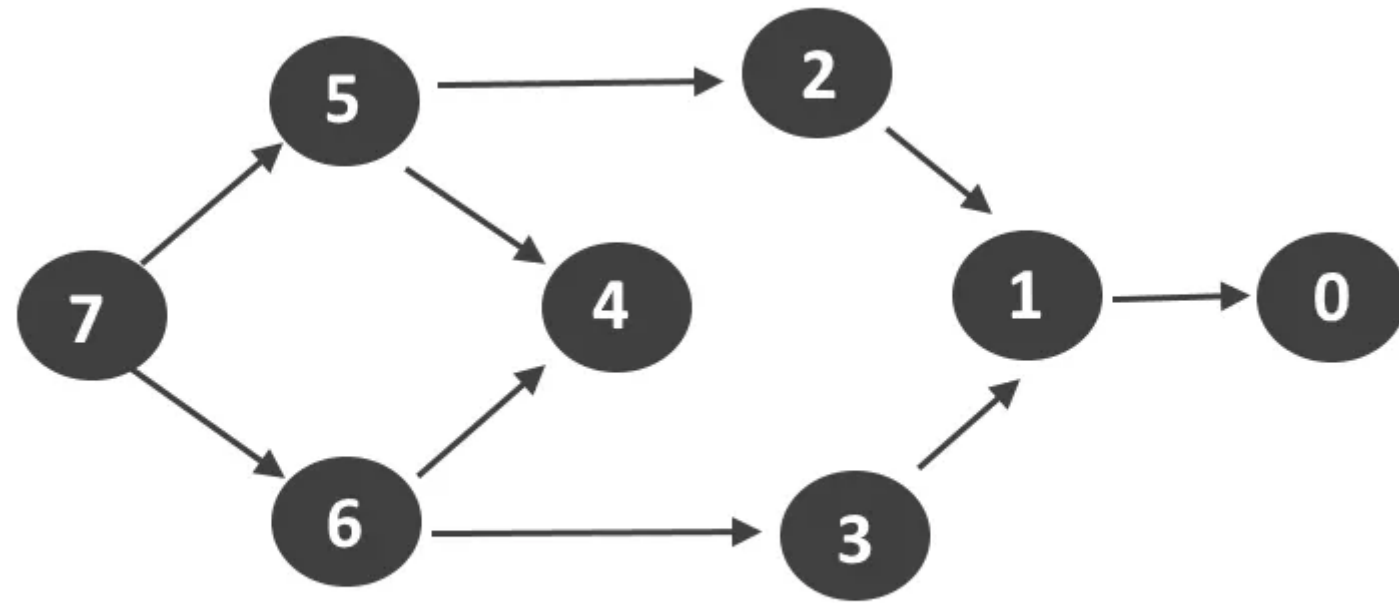
assign the neighbours of the vertices of the previous step the colour red



repeat till all vertices are coloured, or a conflicting colour assignment occurs.

set U: red colour  
set V: blue colour

# Topological sort



Topological Sort : 7 6 5 4 3 2 1 0