Introduction to Graph Theory

Jeff Chak Fu WONG

Department of Mathematics, Chinese University of Hong Kong, Shatin, Hong Kong

EDB HSMMC Math Modelling Advanced Training Workshop

What is a graph?

A graph represents a network which consists of a set of objects, mathematically called **vertices** or **nodes**. These vertices or nodes are interconnected with each other based upon some relation, with the help of nodes or **edges** or **arcs**.

一個**圖**用以表示網絡，其由一組物件構成；該類物件在數學上稱為**頂點**或**節點**。此等頂點或節點會依據特定關係，透過**邊**或**弧**實現彼此連結。

Definition of Graph

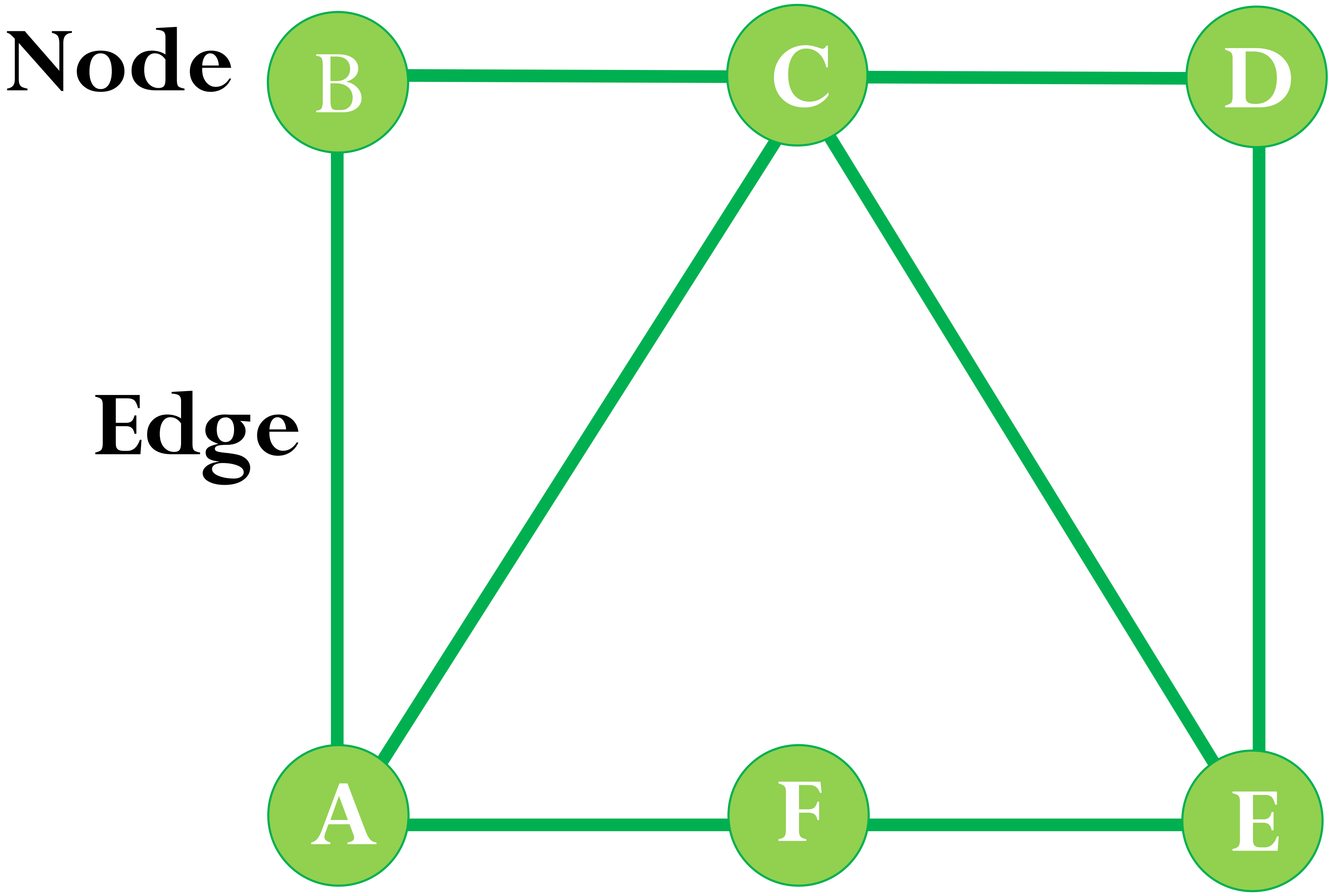
A graph is a mathematical structure used to represent a set of objects and the connections between them:

- The objects are called vertices (or nodes),
- The connections between them are called edges (or links).
  - ***Nodes:*** *A finite non-empty set of points.*
  - ***Edges:*** *A set of pairs of vertices that represent the connections between them.*

圖是一種用於表示一組物件及其彼此間連結關係的數學結構：

- 此類物件稱為**頂點**（或節點）；
- 物件之間的連結稱為**邊**（或連結）。
  - **節點**：由有限個點所構成的非空集合。
  - **邊**：由頂點配對所組成的集合，用以表示頂點之間的連結。

Kruskal's Algorithm



Formally

A graph G is defined as: $G = (V, E)$

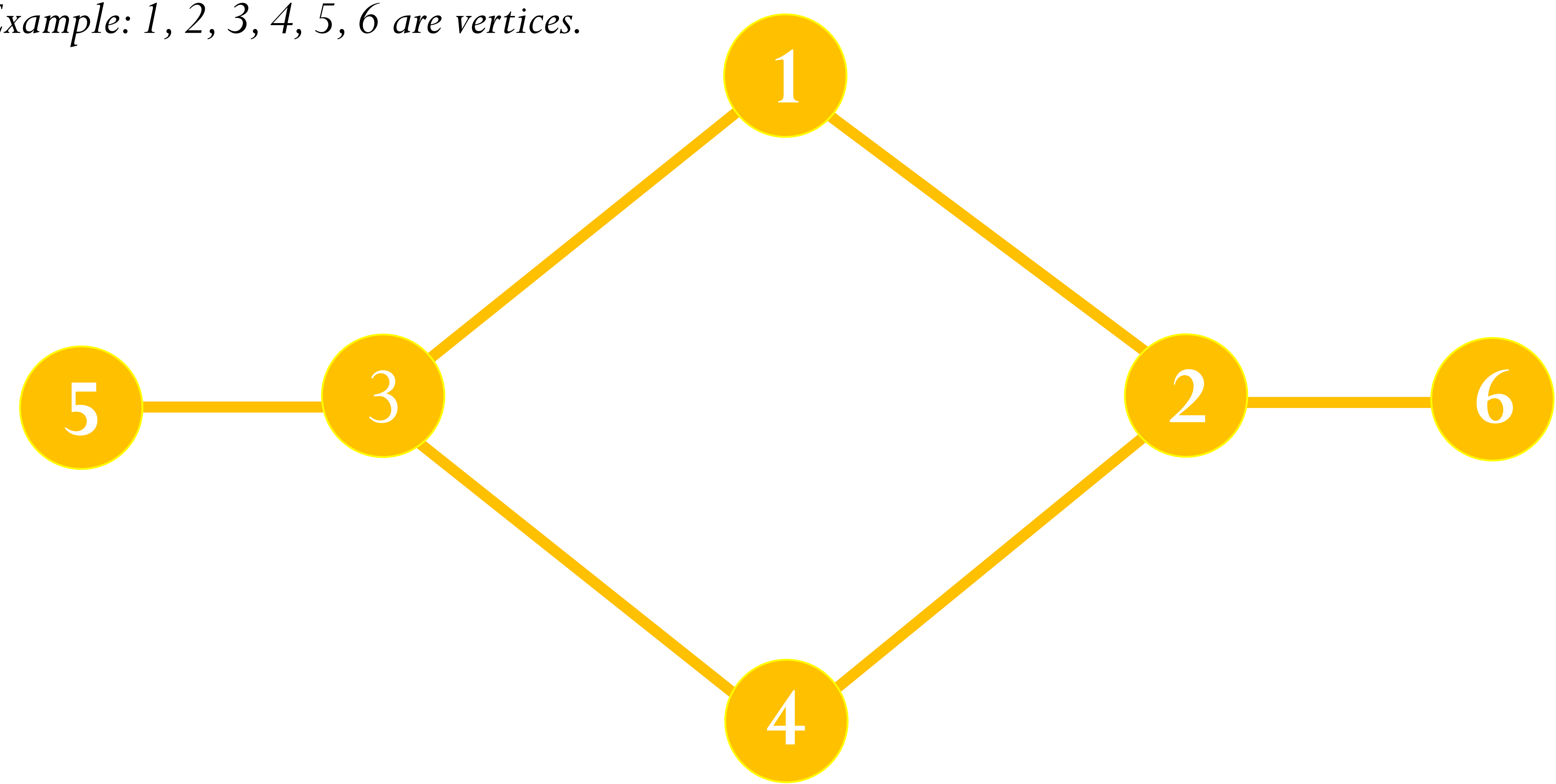
where:

$V =$ a set of **vertices (or nodes)**, representing objects.
$E =$ a set of **edges (or links)**, representing connections between pairs of vertices.

Vertex (Node): A fundamental element of a graph, representing an object, entity, or point.

Vertex (Node): *The circles labeled in the graph.*

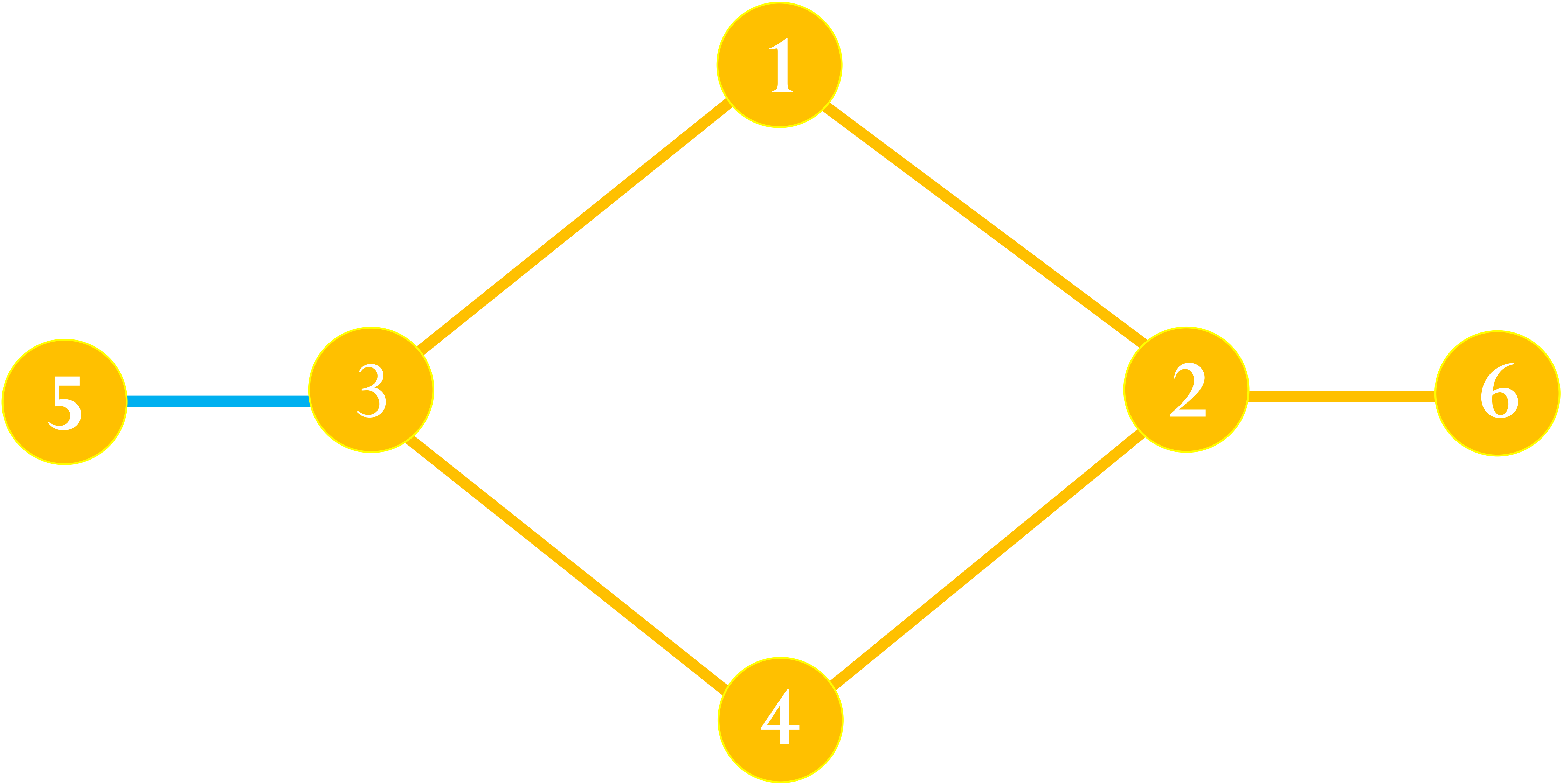
Example: 1, 2, 3, 4, 5, 6 are vertices.



Edge (Link): A connection between two vertices, showing a relationship or pathway.

Edge (Link): *The lines connecting two vertices.*

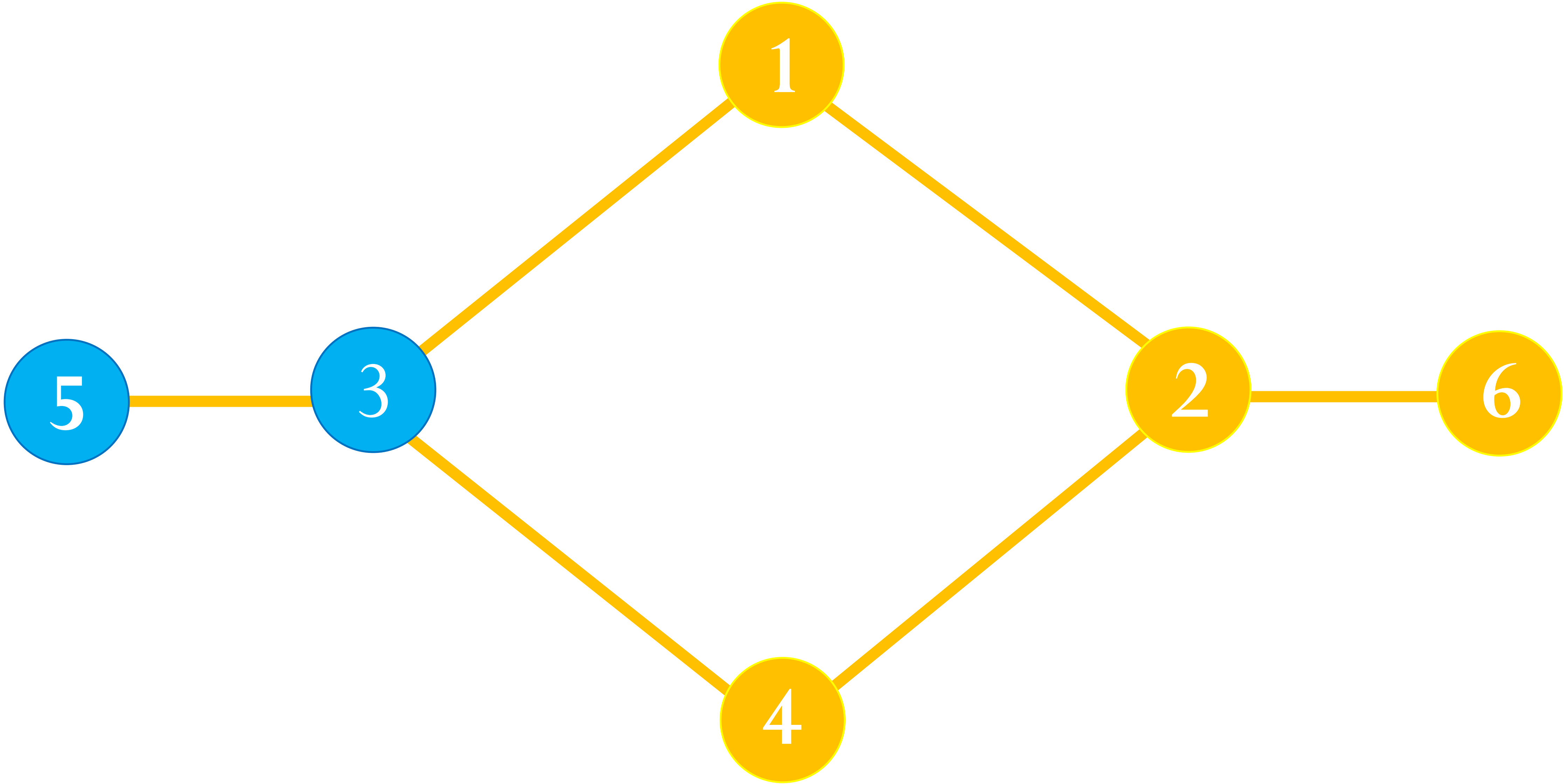
Example: The line between vertex 3 and 5 is an edge.



Adjacent Vertices: Two vertices that are directly connected by an edge.

Adjacent Vertices: *Vertices that are directly connected by an edge.*

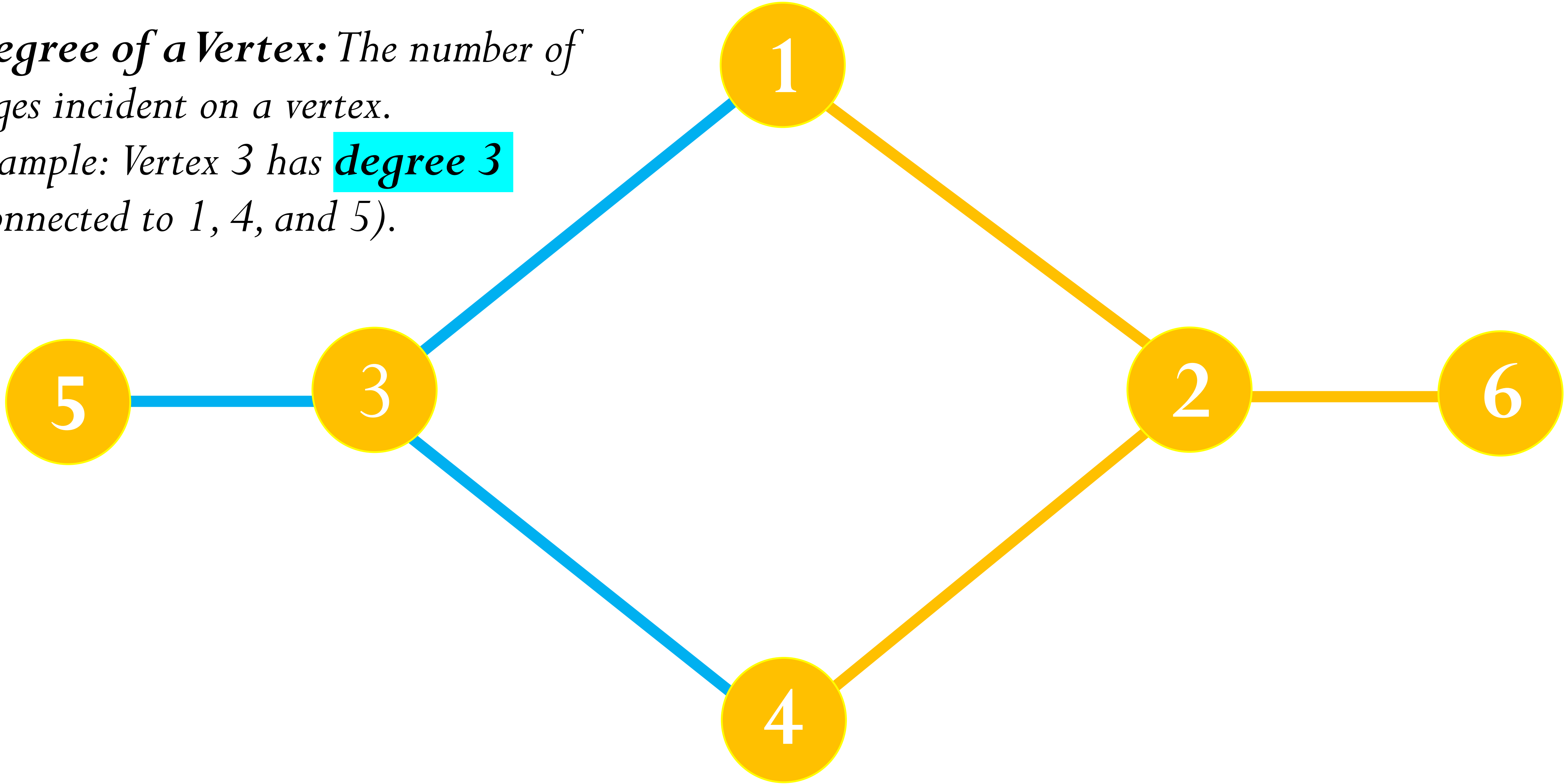
Example: 3 and 5 are adjacent because they share an edge.



Degree of a Vertex: The number of edges incident on a vertex.

- In a **directed graph**, we distinguish between **in-degree** (incoming edges) and **out-degree** (outgoing edges).

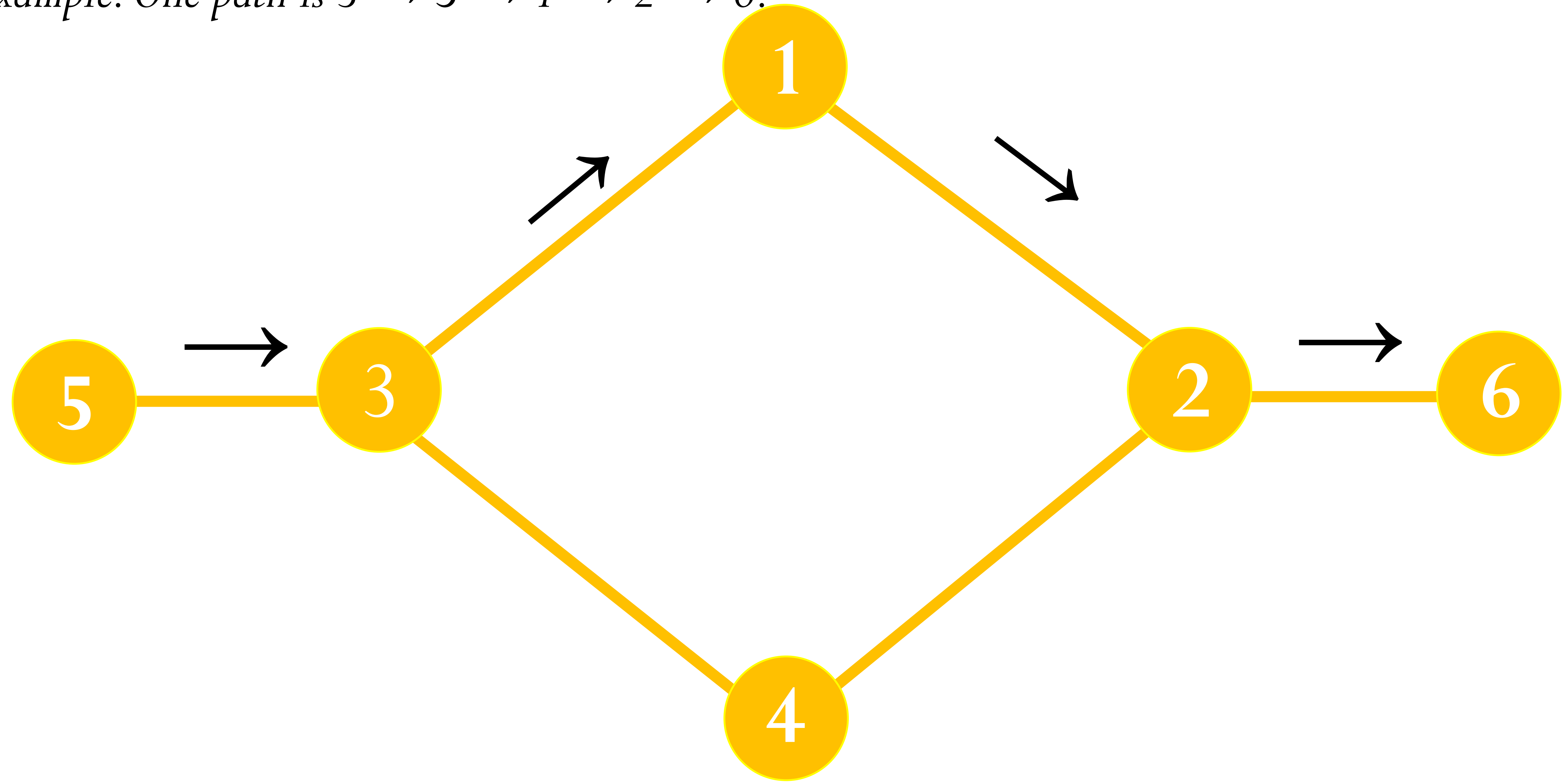
Degree of a Vertex: *The number of edges incident on a vertex.*
Example: Vertex 3 has ***degree 3*** *(connected to 1, 4, and 5).*



Path: A sequence of vertices connected by edges, with no vertex repeated.

Path: *A sequence of vertices connected by edges.*

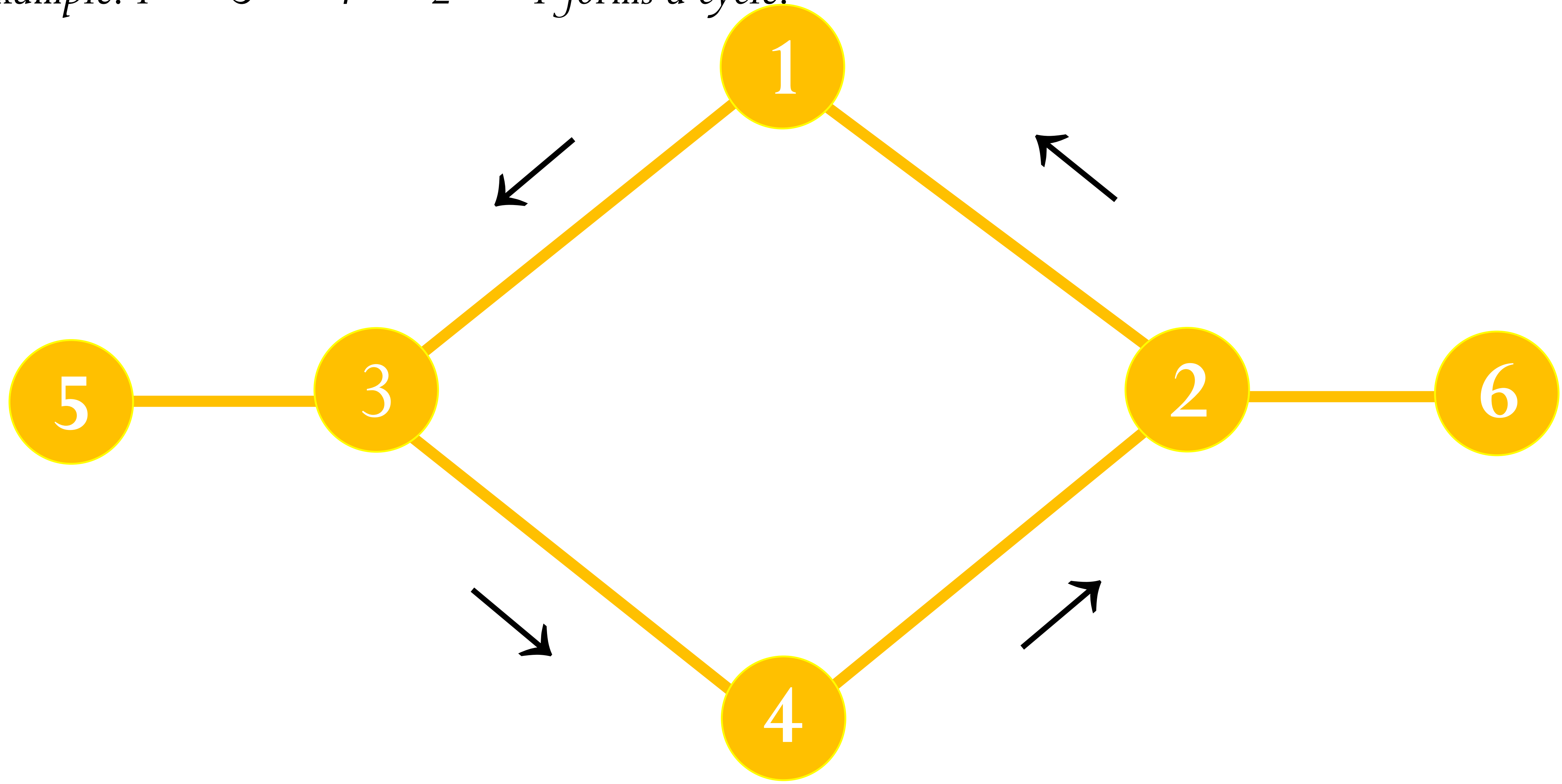
Example: One path is 5 ⟶ 3 ⟶ 1 ⟶ 2 ⟶ 6.



Cycle: A path that begins and ends at the same vertex, forming a closed loop.

Cycle: *A path that starts and ends at the same vertex, without repeating edges.*

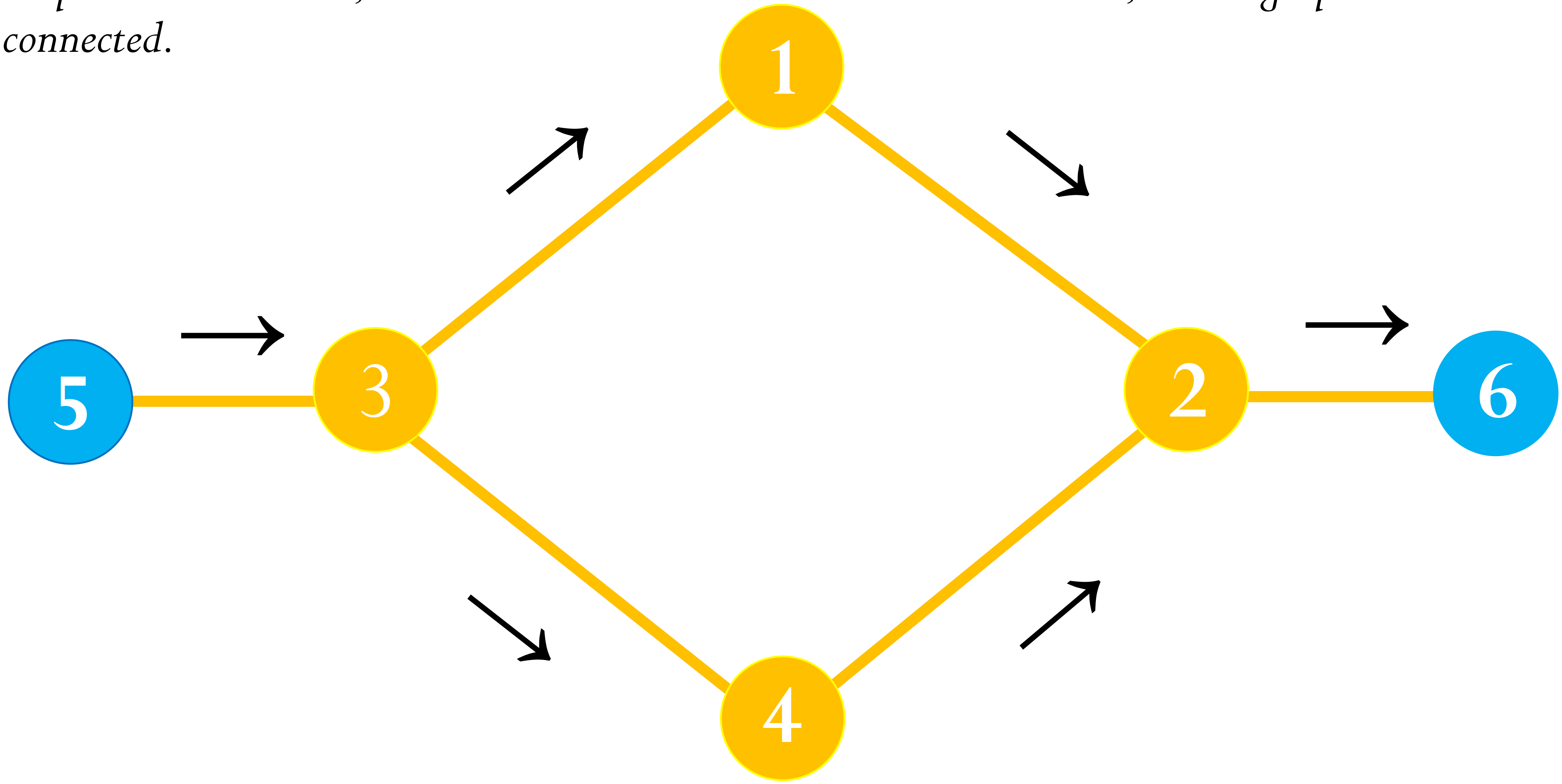
Example: $1 \longrightarrow 3 \longrightarrow 4 \longrightarrow 2 \longrightarrow 1$ *forms a cycle.*



Connected Graph: A graph in which there exists a path between every pair of vertices.

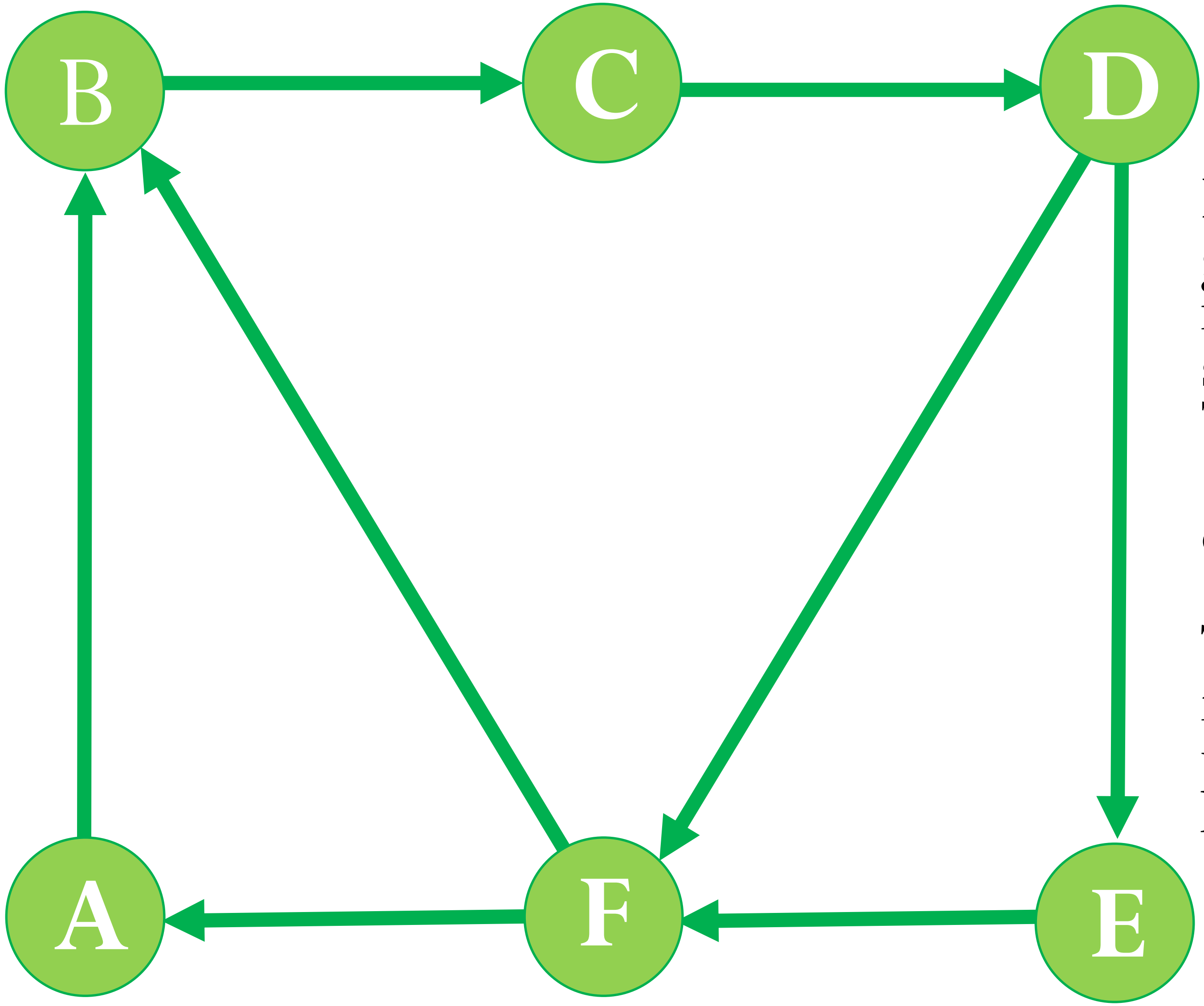
Connected Graph: *A graph where there is a path between every pair of vertices.*

Example: From vertex 5, we can reach vertex 6 via 5 ⟶ 3 ⟶ 2 ⟶ 6, so this graph is connected.



Directed Graph

A graph in which the direction of the edge is defined for a particular node is a directed graph.

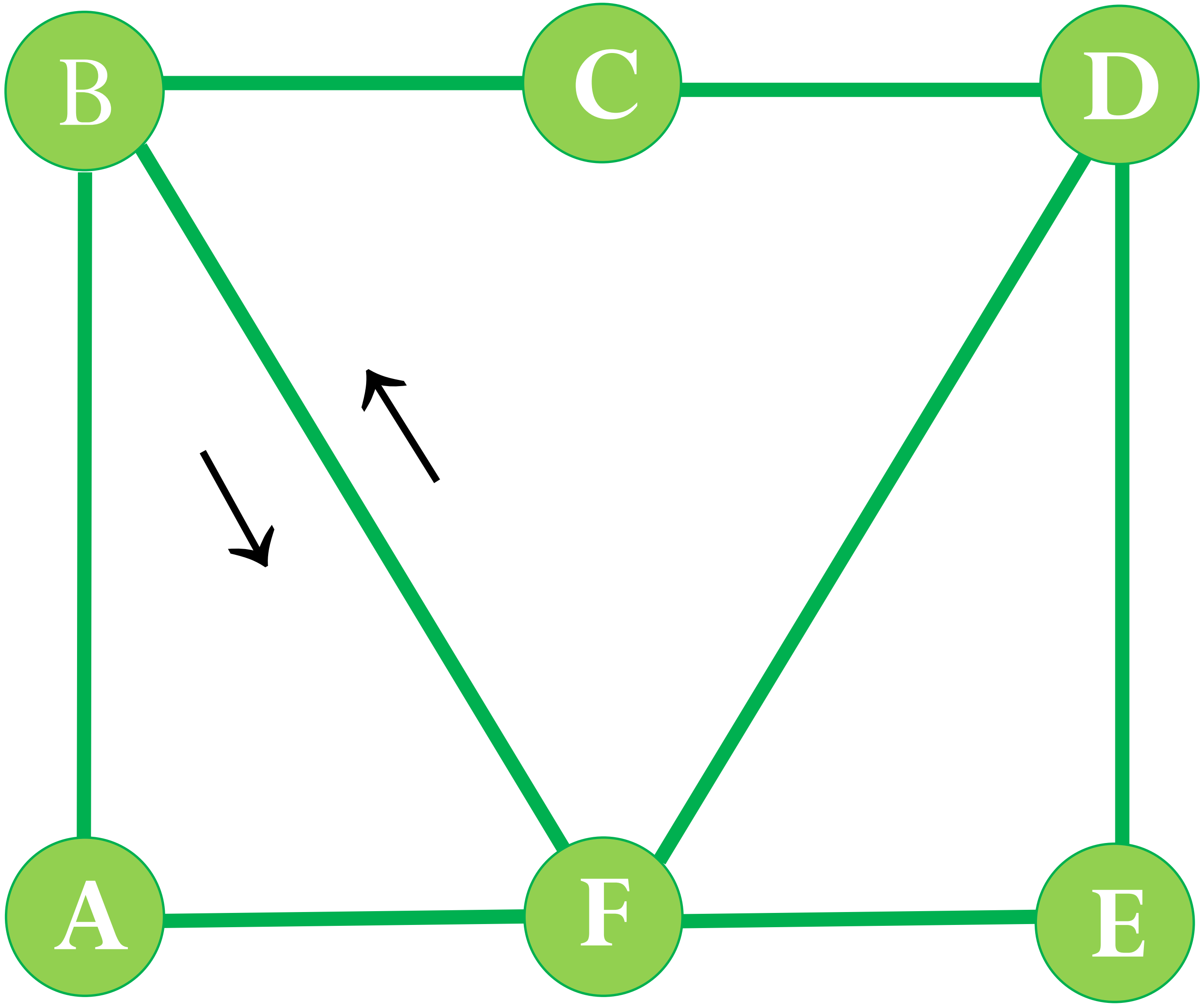


Directed Acyclic: It is a directed graph with no cycle. For a vertex '*v*' in DAG, there is no directed edge starting and ending with vertex '*v*'. The arrows go in one direction only (Directed), and you can't go in a circle or loop (Acyclic).

Tree: A tree is just a restricted form of graph. That is, it is a DAG with a restriction that a child can have only one parent.

Undirected Graph

An undirected graph in which the direction of the edge is not defined. So, if an edge exists between node '*u*' and '*v*', then there is a path from node '*u*' to '*v*' and vice versa.



Representation of Graph

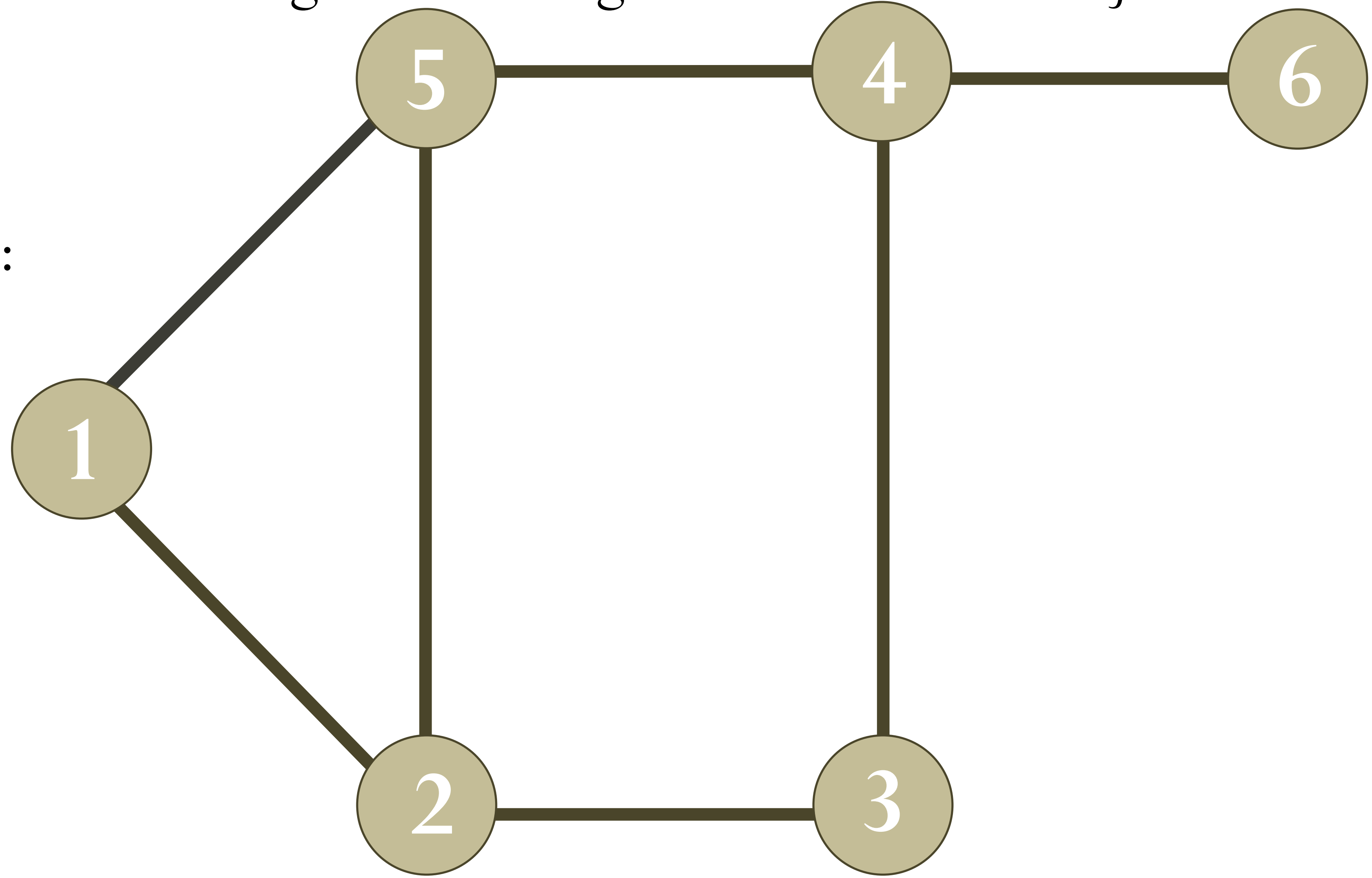
Here are the two most common ways to represent a graph : For simplicity, we are going to consider only unweighted graphs in this post.

- Adjacency Matrix
- Adjacency List

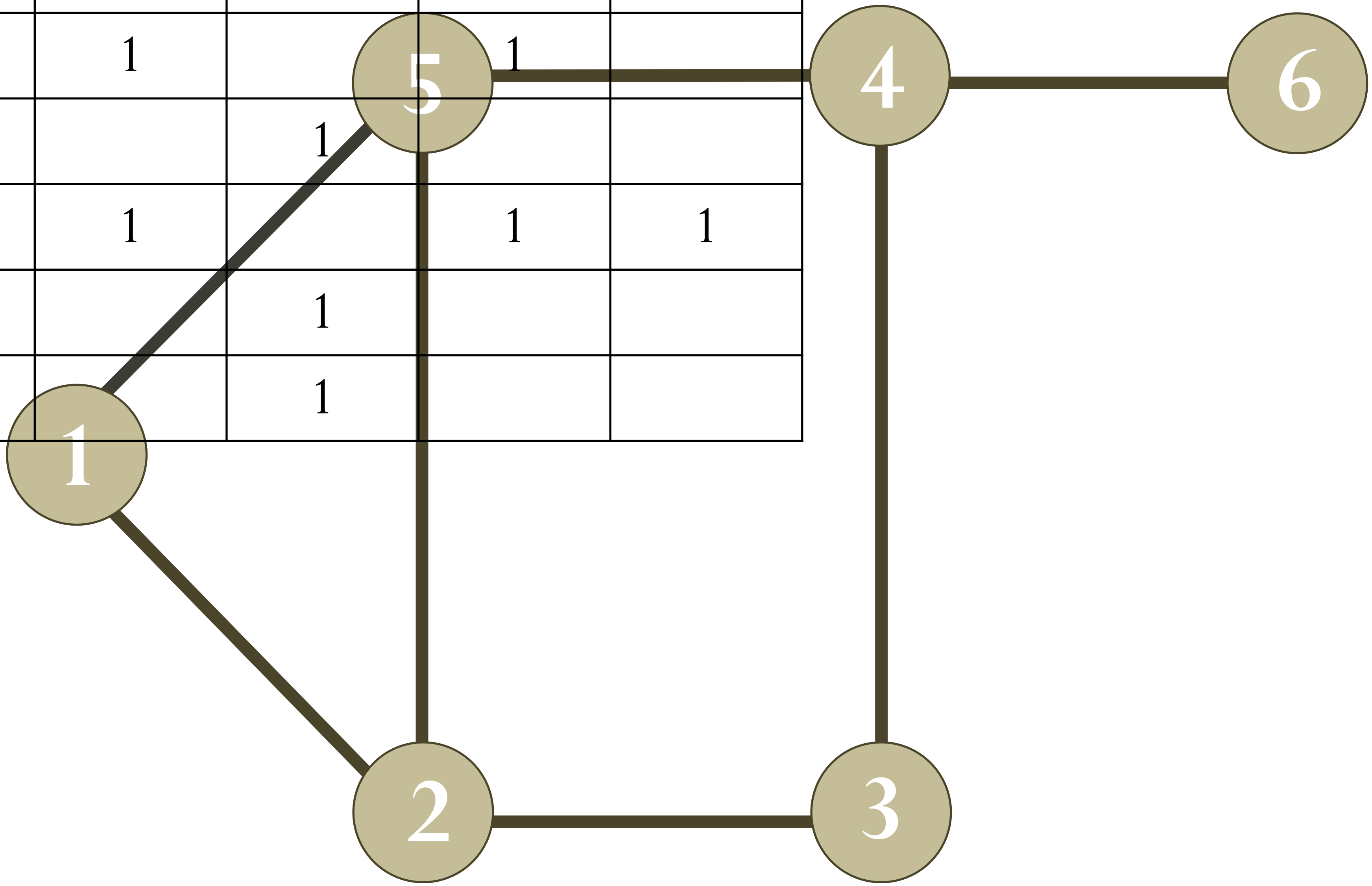
Adjacency Matrix Representation

If $M_{ij} = 1$, it means there is an edge connecting vertex i and vertex j;
if $M_{ij} = 0$, it means there is no edge connecting vertex i and vertex j.

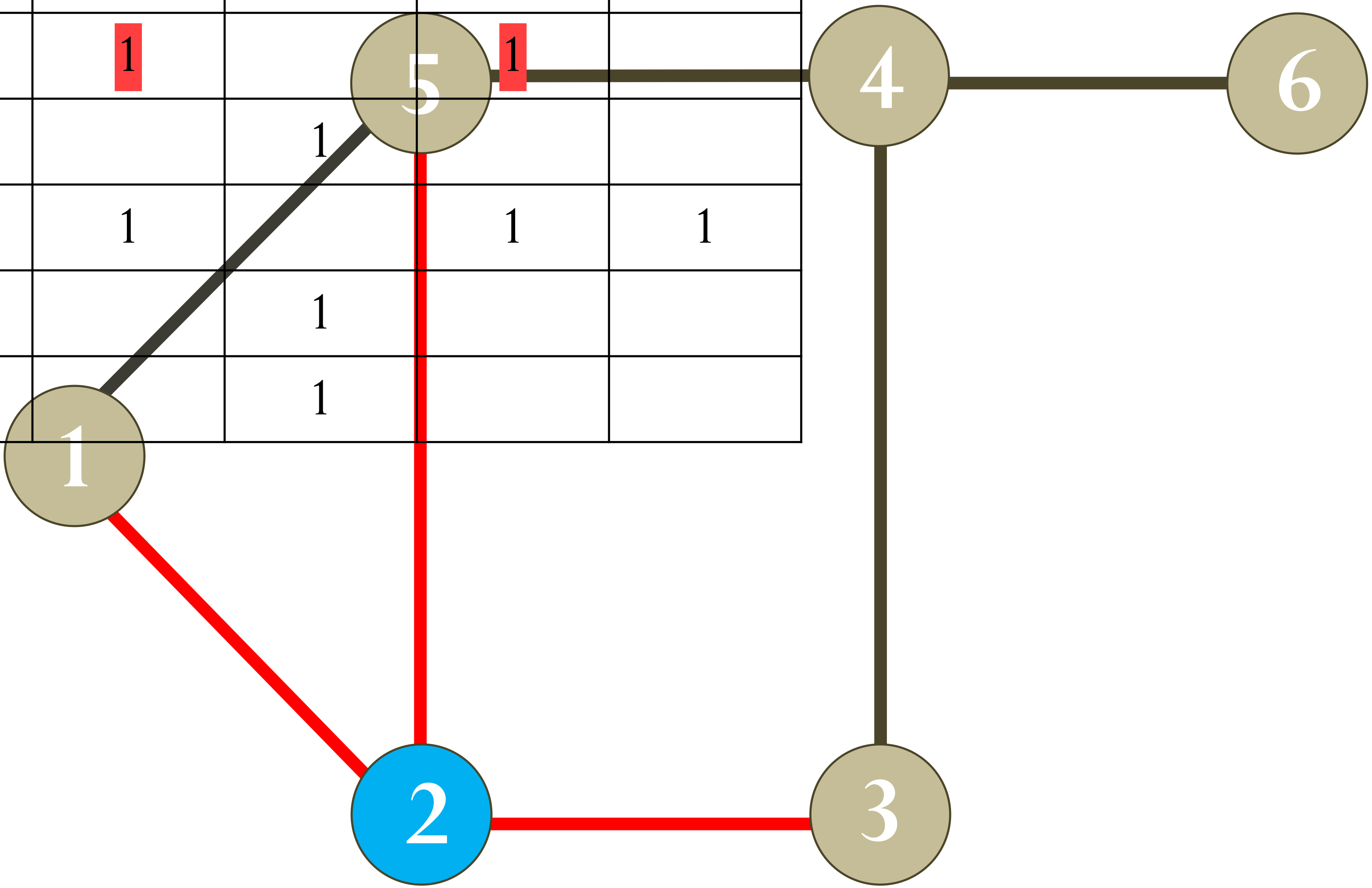
Let us consider the following 6x6 matrix:



	1	*2*	*3*	*4*	*5*	*6*
1		1			1	
2	1		1		1	
3		1		1		
4			1		1	1
5	1	1		1		
6				1		



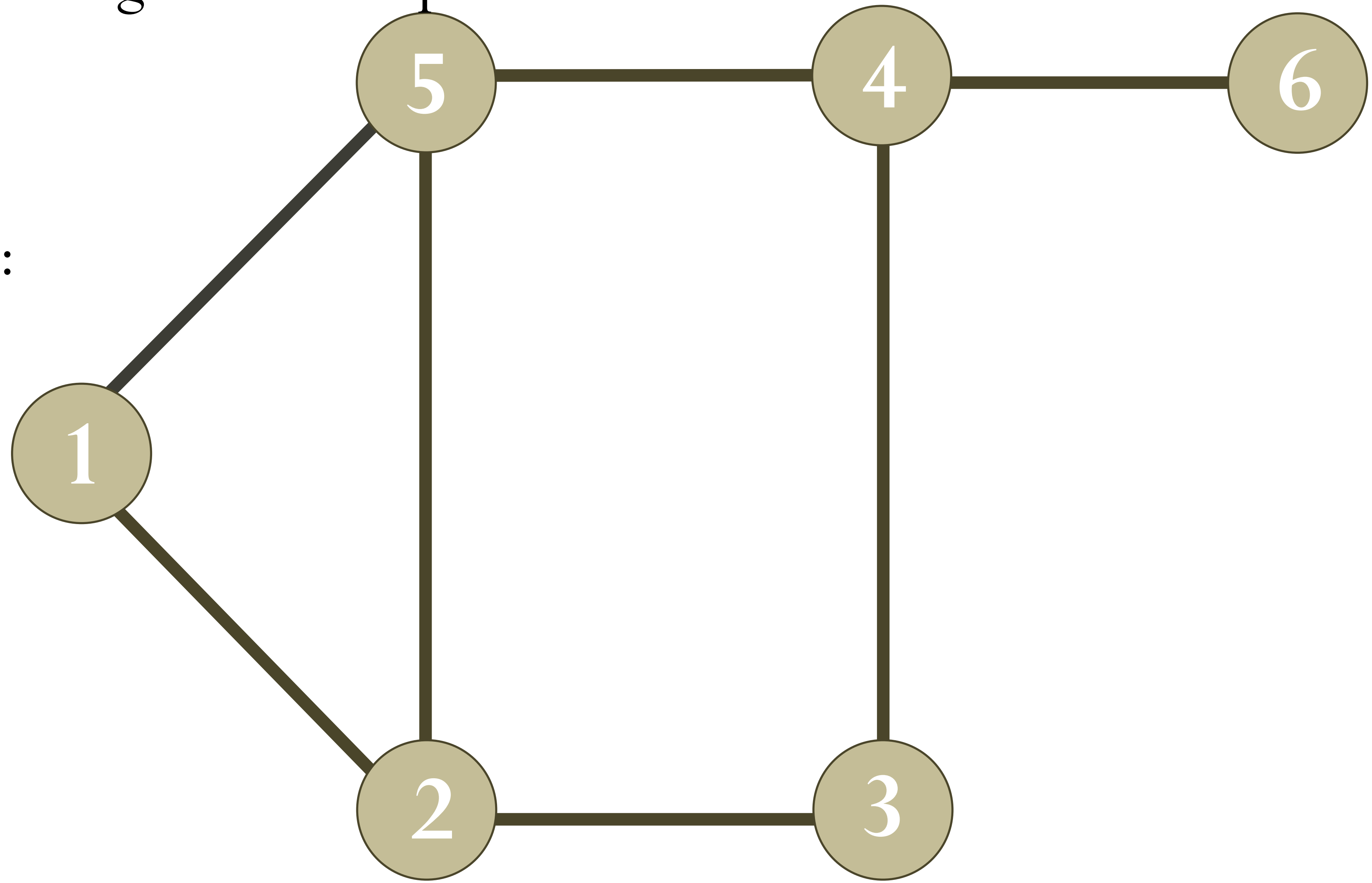
	1	*2*	*3*	*4*	*5*	*6*
1		1			1	
2	1		1		1	
3		1		1		
4			1		1	1
5	1	1		1		
6				1		



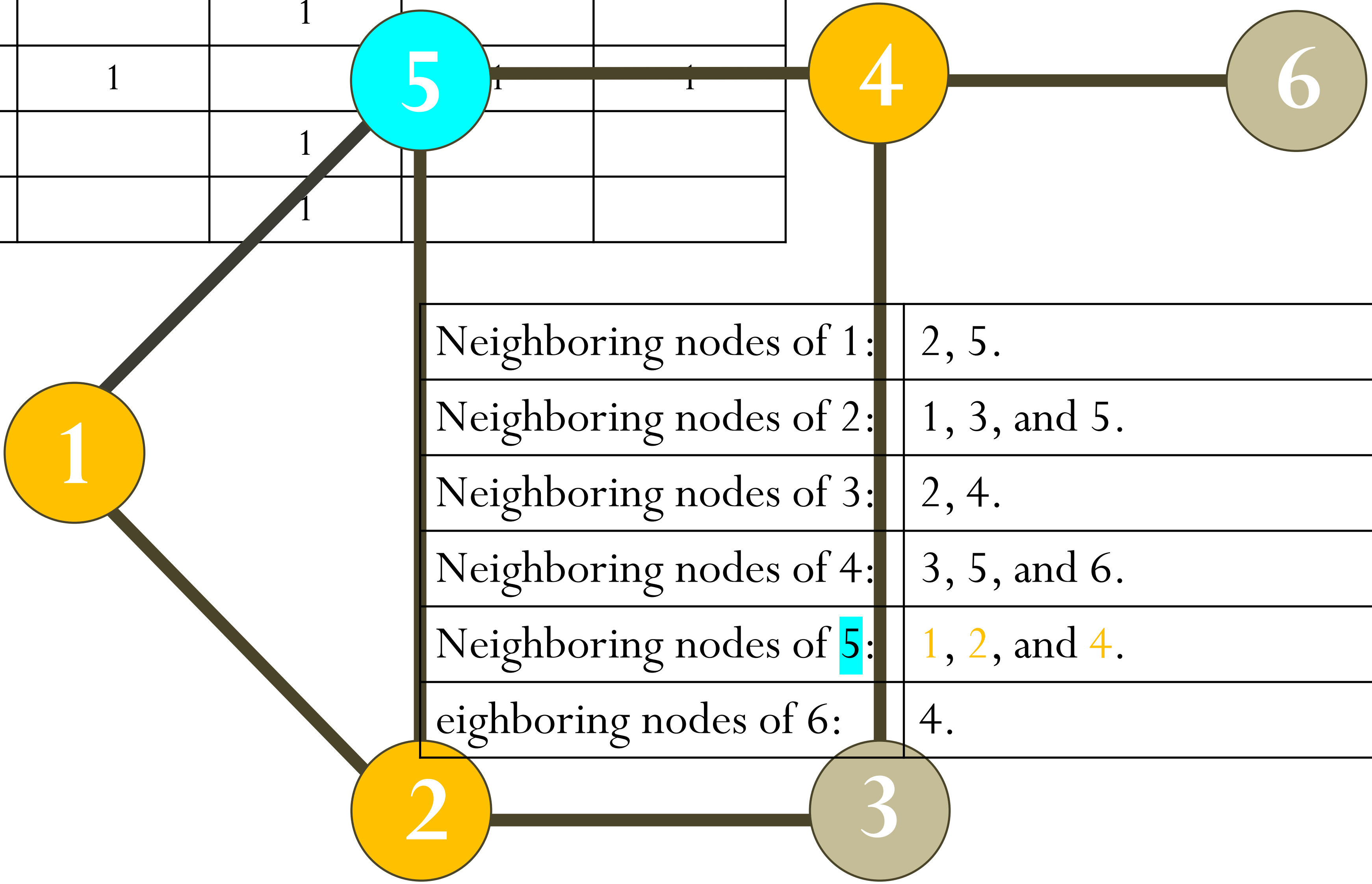
Adjacency list

In this case, all the zeroes of the adjacency matrix are eliminated and only the corresponding neighboring nodes of a particular node are considered.

Let us consider the following 6x6 matrix:



	1	*2*	*3*	*4*	*5*	*6*
1		1			1	
2	1		1		1	
3		1		1		
4			1		1	1
5	1	1		1		
6				1		



Neighboring nodes of 1:	2, 5.
Neighboring nodes of 2:	1, 3, and 5.
Neighboring nodes of 3:	2, 4.
Neighboring nodes of 4:	3, 5, and 6.
Neighboring nodes of 5:	1, 2, and 4.
eighboring nodes of 6:	4.

Question 6