CIS 606 Analysis of Algorithms

Elementary Graph Algorithms

RATIONALE

- Graphs are a powerful means of modeling data in real-world such as social median networks, web pages and links, road maps in a GIS system.
- Graph problems pervade computer science and hundreds of interesting computational problems are couched in terms of graphs, e.g., computing the shortest path.
- Developing graph algorithms is fundamental to computer science.

OBJECTIVES

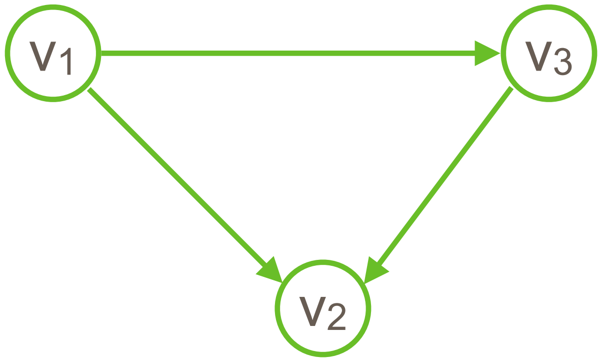
- Understand the definition of a graph
- Understand different type of graphs
- Learn to use adjacent list and matrix to represent a graph in the computer system

PRIOR KNOWLEDGE

- Linked list
- Matrix

GRAPH

- A graph is a structure linking a set of objects.
- A graph is a pair $G=(V, E)$:
  - V is a set of vertices, known as nodes, $V=\{v_1, v_2, \ldots, v_n\}$;
  - E is a set of edges, $E = \{e_1, e_2, \ldots, e_n\}$, where each edge e_i is a pair of vertices (v_i, v_j) and connects two vertices v_i and v_j.
-

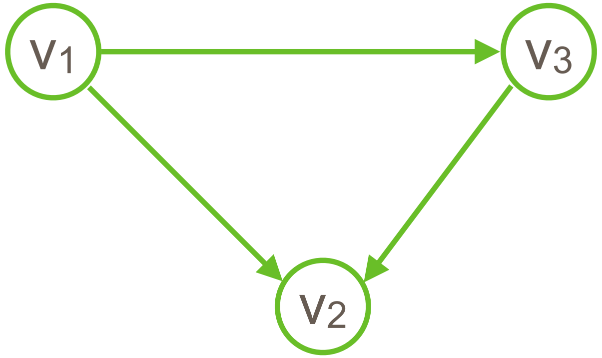


$V = \{v_1, v_2, v_3\}$

$E = \{(v_1, v_2), (v_1, v_3), (v_3, v_2)\}$

DIRECTED GRAPHS

- In a directed graph (also called digraph), every edge has a direction.
- For edge $e_i = (v_i, v_j)$, v_i is the source and v_j is the destination.
- In-degree of a vertex v is the number of edges coming toward to v.
- Out-degree of a vertex v is the number of outgoing edges



$V = \{v_1, v_2, v_3\}$

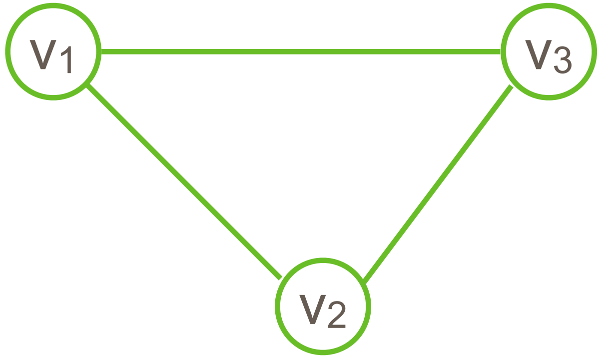
$E = \{(v_1, v_2), (v_1, v_3), (v_3, v_2)\}$

$\text{In-degree}(v_1) = 0$

$\text{Out-degree}(v_1) = 2$

UNDIRECTED GRAPH

- In an undirected graph, edges have no specific directions or always “two-way”.
- Degree of a vertex is the number of edges connecting that vertex or the number of adjacent vertices.



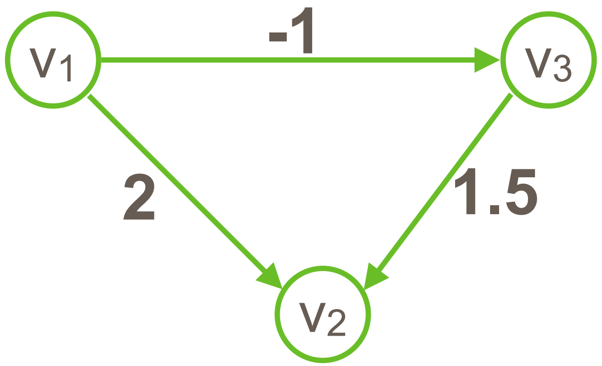
Degree(v_2) = 2

$$V = \{v_1, v_2, v_3\}$$

$$E = \{(v_1, v_2), (v_1, v_3), (v_2, v_3), (v_2, v_1), (v_3, v_1), (v_3, v_2)\}$$

WEIGHTED GRAPH

- In a weighted graph, every edge $e_i = (v_i, v_j)$ has a weight/cost $w(v_i, v_j)$



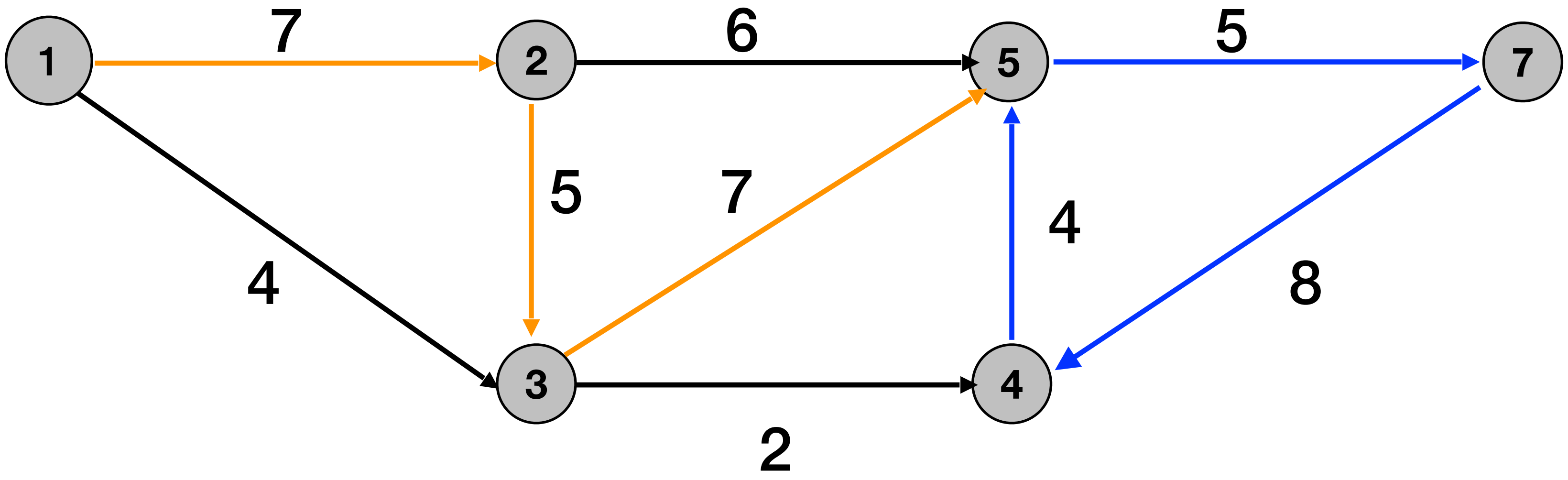
$w(v_1, v_2) = 2$

$w(v_1, v_3) = -1$

$w(v_3, v_2) = 1.5$

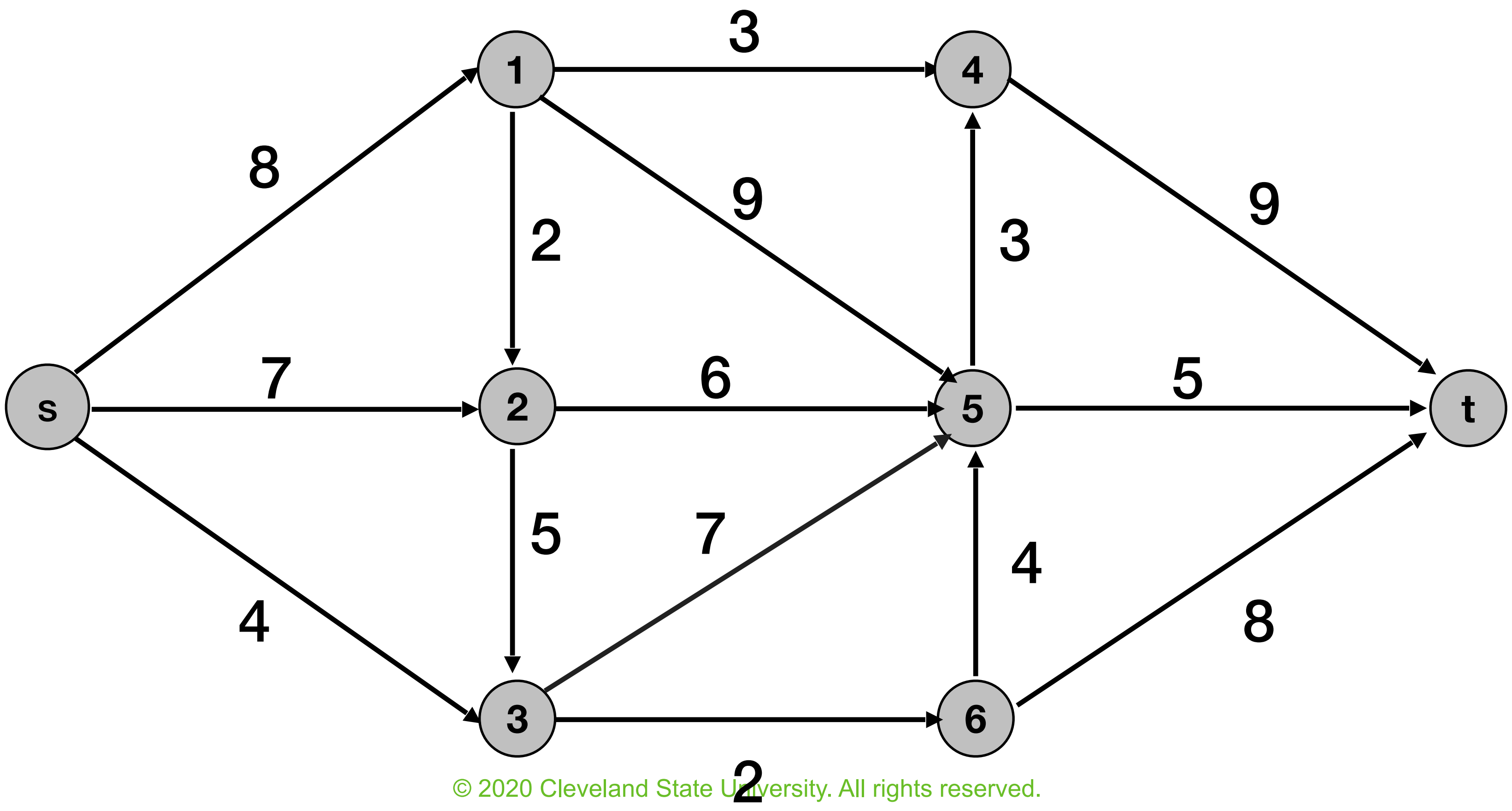
PATH

- A path on a graph $G=(V, E)$ is a list of vertices $\{v_0, v_1, \ldots, v_k\}$ such that (v_i, v_j) is an edge in E for all $0 <= i <= k$, and we say a path from v_0 to v_k.
- A cycle is a path that begins and ends at the same node.



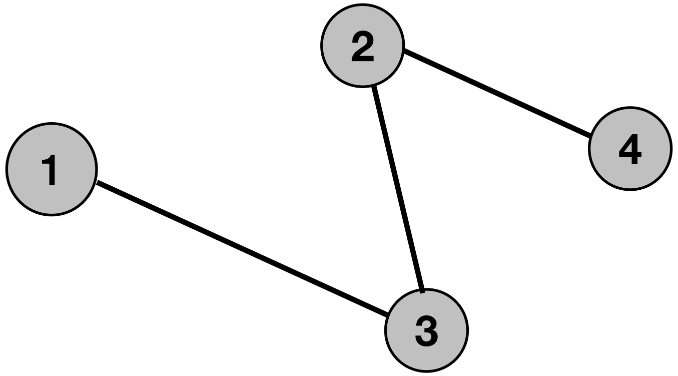
DIRECTED ACYCLIC GRAPHS (DAGS)

- A DAG is a directed acyclic graph without cycles.

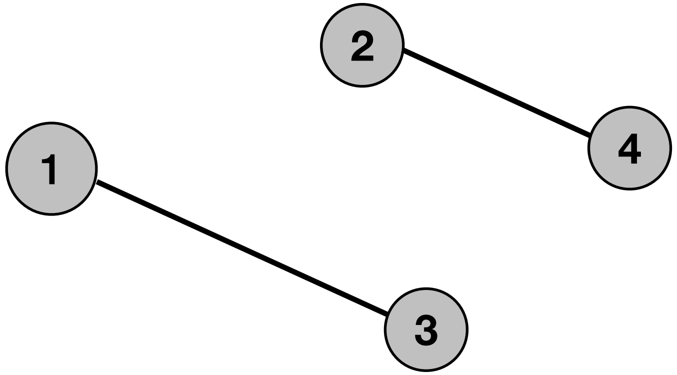


GRAPH CONNECTIVITY

- A undirected graph is connected if for all pairs of vertices u and v, there exists a path from u to v.



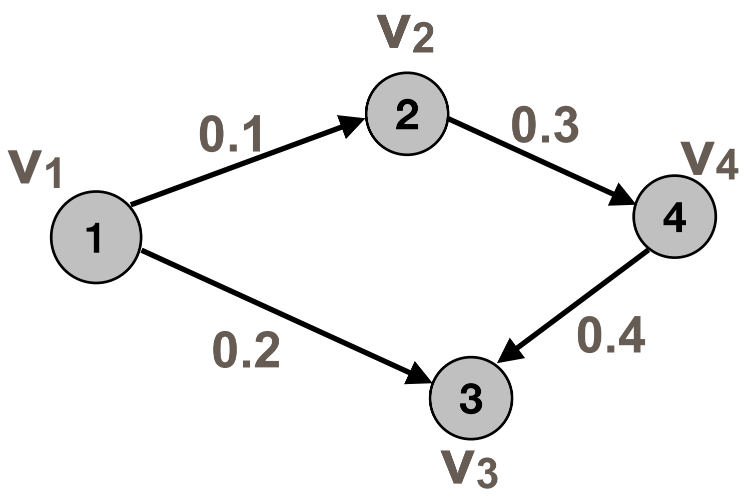
connected

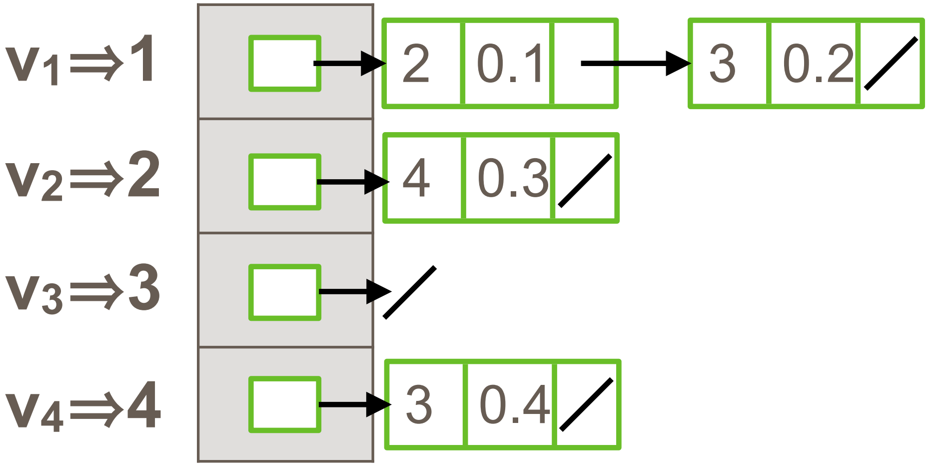


disconnected

GRAPH REPRESENTATION—ADJACENCY LISTS

- Adjacent Lists — G = (V, E) where |V| = n and |E| = m
  - Assign each node a number from 1 to n
  - An array of length n in which each entry stores a list of all adjacent vertices for a vertex in V.

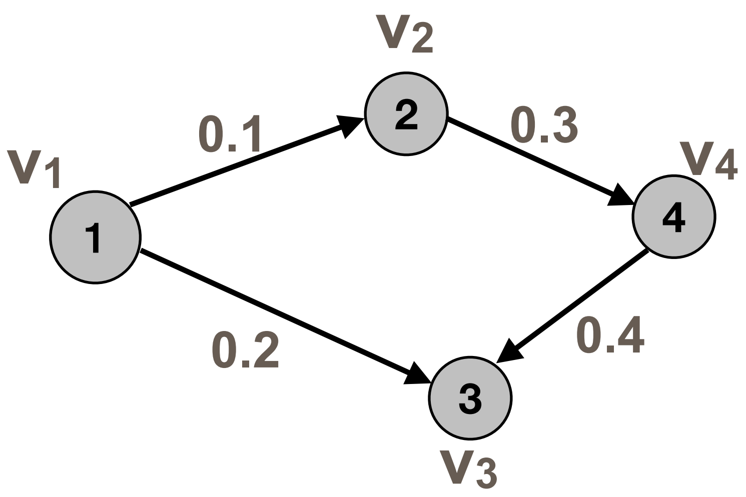


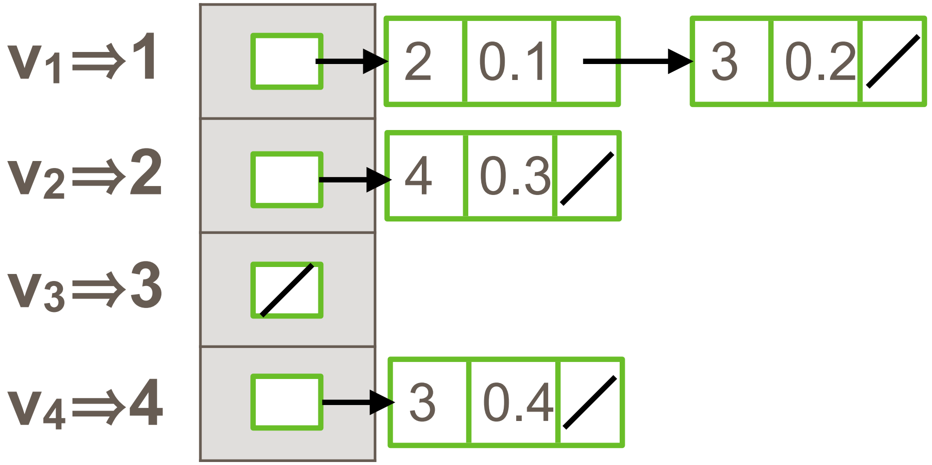


ADJACENCY LIST PROPERTY

Operations	Time
Out-degree(v)	O(m)
In-degree(v)	O(n+m)
Exist(e=(u,v))	O(m)
Insert(e)	O(1)
Delete(e)	O(m)

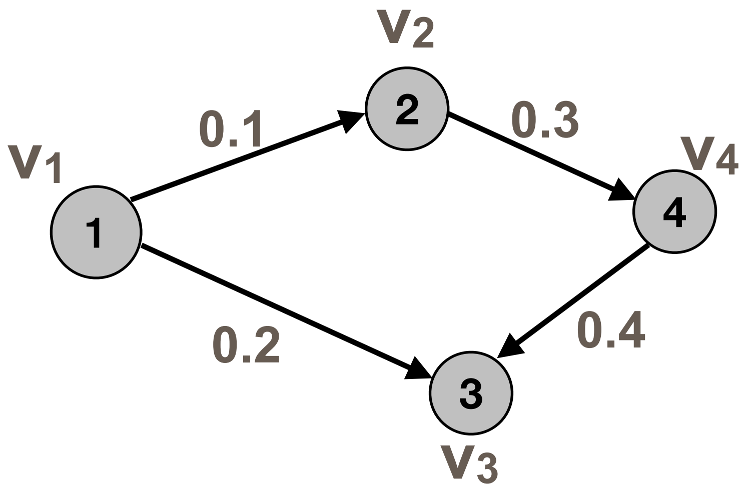
Space: O(|V|+|E|) = O(n+m)





GRAPH REPRESENTATION-ADJACENCY MATRIX

- Adjacent Lists — G = (V, E) where |V| = n and |E| = m
  - Assign each node a number from 1 to n
  - A n×n matrix M where M[i, j] = the weight of e=(v_i, v_j) if e exists, otherwise M[i, j]=0.

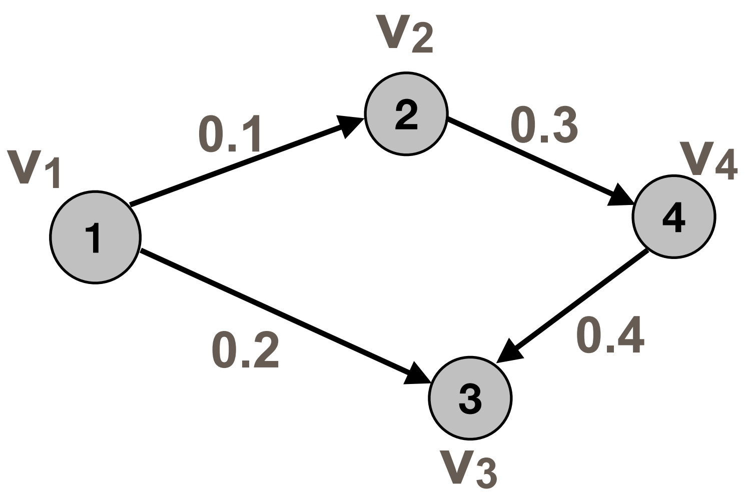


	v_1 1	v_2 2	v_3 3	v_4 4
v_1 1	0	0.1	0.2	0
v_2 2	0	0	0	0.3
v_3 3	0	0	0	0
v_4 4	0	0	0.4	0

ADJACENCY MATRIX PROPERTY

Operations	Time
Out-degree(v)	O(n)
In-degree(v)	O(n)
Existence(e=(u,v))	O(1)
Insert(e)	O(1)
Delete(e)	O(1)

Space: $O(|V|\times|V|) = O(n^2)$



	1	2	3	4
1	0	0.1	0.2	0
2	0	0	0	0.3
3	0	0	0	0
4	0	0	0.4	0

SUMMARY

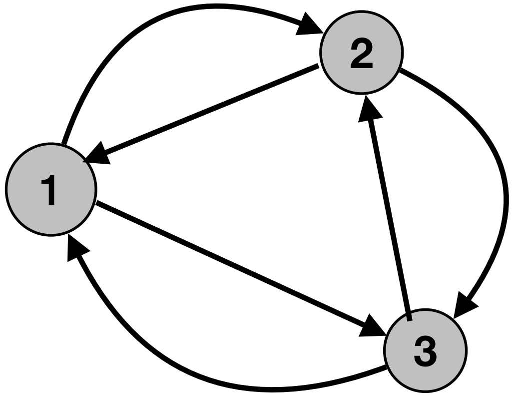
- A graph consist of a set V of vertices and a set E of edges.
- Every edge connect two vertices and may have a direction.
- A path from v to u on a graph is a sequence of vertices from v to u such that every two adjacent vertex has an edge.
- A graph is connected if for all pairs of vertices, there is a path from u to v.
- There are two representations for a graph: adjacency list and adjacency matrix.
- In the adjacency list, every vertex u has a list to store all its adjacent vertices.
- In the adjacency matrix, every edge (u,v) corresponds a cell in it.

TO PREPARE FOR THE NEXT LESSON

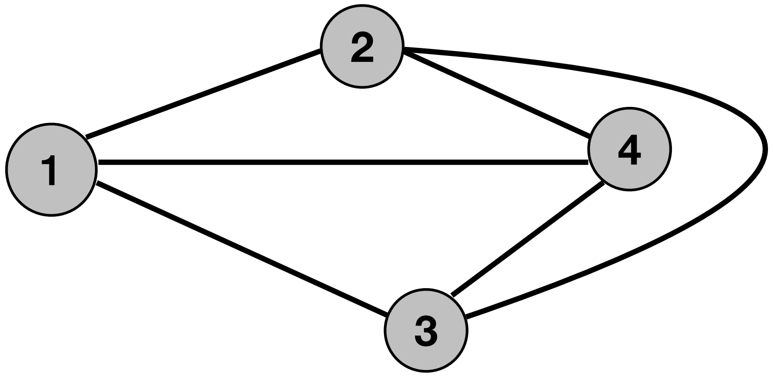
- Read the Chapter 22.2 for BFS.

GRAPH CONNECTIVITY (CONT)

- **A graph is fully connected/complete if there exists an edge from every vertex to every other vertex.**



Fully connected



Fully connected