Graphs

Graphs vs. trees

- A **graph** is a set of nodes (a.k.a. **vertices**) that may be connected by **edges**
- You can think of a graph as a generalization of a tree
  - In a tree, each node has one "previous" node and any number of "next" nodes
  - In a graph, each node can be connected to any number of nodes (including itself!) There is no notion of "hierarchy" like there is with a tree.

Types of graphs

- A **simple graph** is one in which each pair of vertices is connected by at most one edge
- In a **multigraph**, each pair of vertices may be connected by more than one edge
- A **weighted graph** assigns numerical values to its edges (these might represent cost, length, or some other metric associated with the edges)

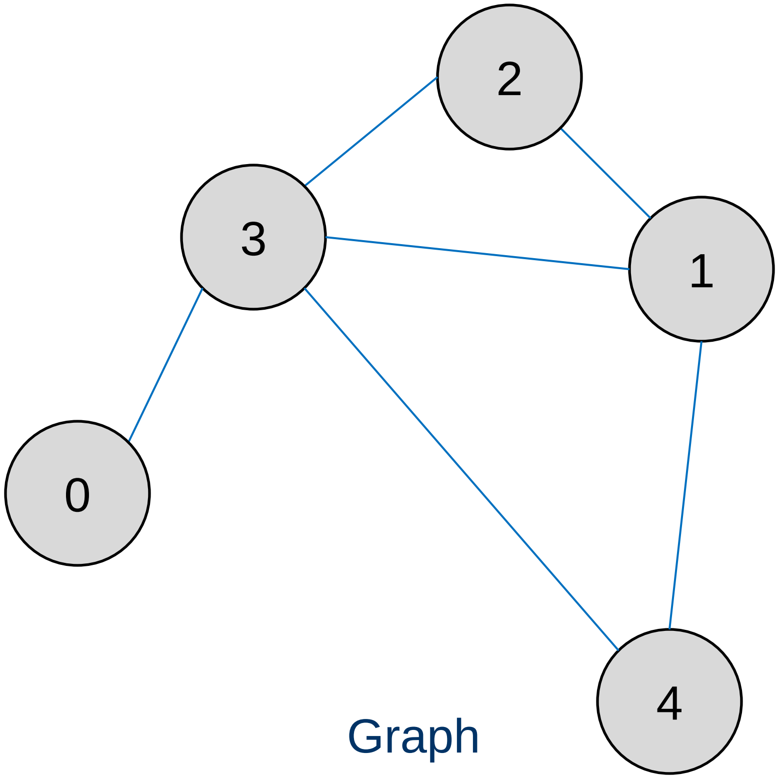
Types of graphs

- In an **undirected graph**, the edges “go both ways” – if there’s an edge between vertices *A* and *B*, it means you can go from *A* to *B* and also from *B* to *A*
- In a **directed graph** (a.k.a. **digraph**), each edge has a specific direction – for example, it’s possible to have an edge that goes from *A* to *B*, but <u>not</u> from *B* to *A*

Graph representations

- Two common computer representations:
  - An **adjacency list** stores a graph with *V* vertices as an array of *V* lists. Each list in the array corresponds to one vertex in the graph. The list associated with vertex *i* stores all the edges that are incident to ("head out of") vertex *i*.
  - An **adjacency matrix** stores a graph with V vertices as a *V* x *V* square matrix. Element [*i*][*j*] in the matrix indicates whether an edge exists between vertices *i* and *j* (1 = edge, 0 = no edge).

Adjacency list

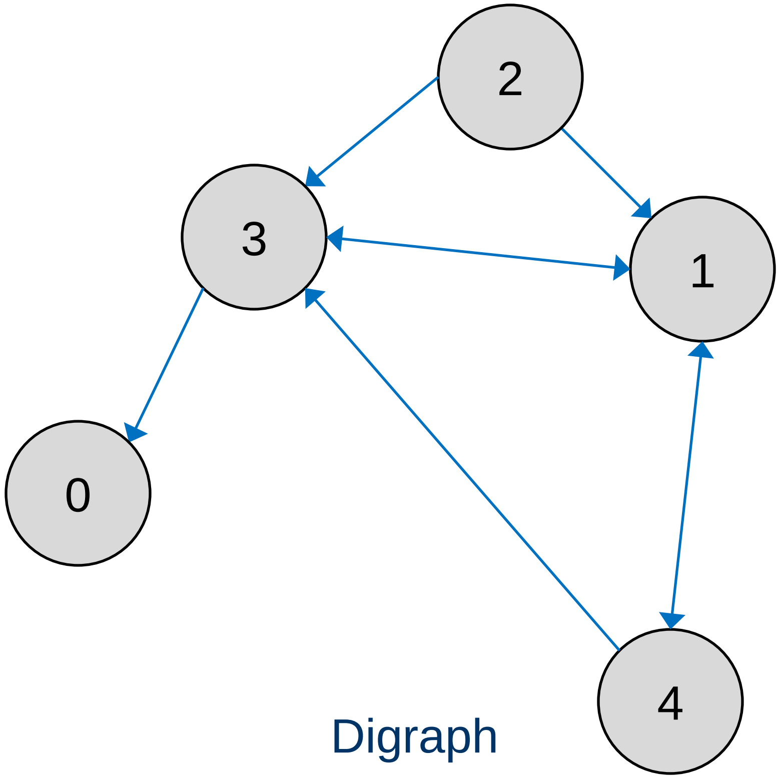


Graph

[0]	0-3
[1]	1-2, 1-3, 1-4
[2]	2-1, 2-3
[3]	3-0, 3-1, 3-2, 3-4
[4]	4-1, 4-3

Equivalent adjacency list
(the notation *i*-*j* indicates the edge between vertices *i* and *j*)

Adjacency list with a digraph

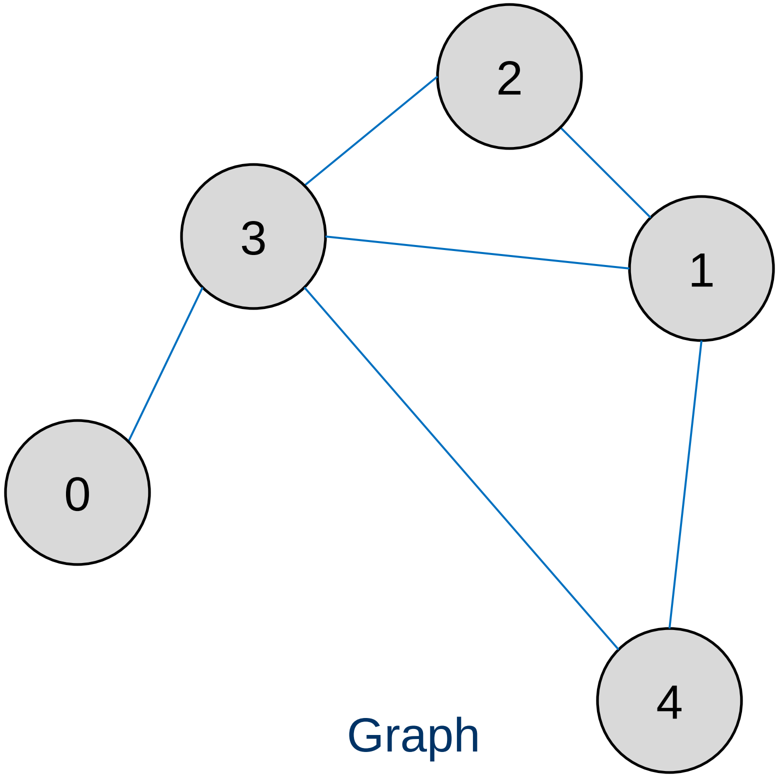


Digraph

[0]	(empty)
[1]	1-3, 1-4
[2]	2-1, 2-3
[3]	3-0, 3-1
[4]	4-1, 4-3

Equivalent adjacency list

Adjacency matrix



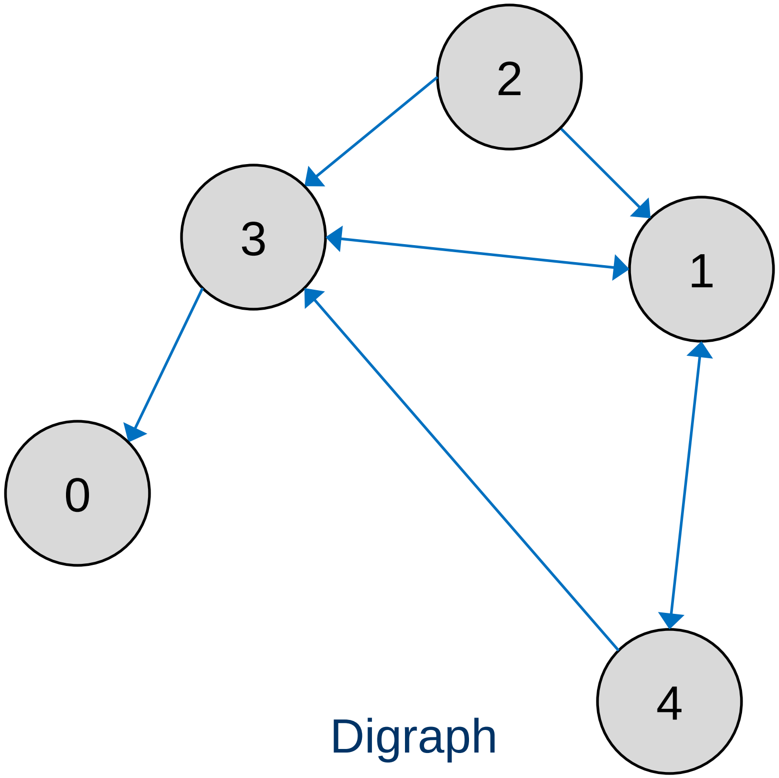
Graph

	[0]	[1]	[2]	[3]	[4]
[0]	0	0	0	1	0
[1]	0	0	1	1	1
[2]	0	1	0	1	0
[3]	1	1	1	0	1
[4]	0	1	0	1	0

Equivalent adjacency matrix

Note that in an undirected graph, the adjacency matrix *A* is symmetric since an edge between vertices *i* and *j* (*A*[*i*][*j*] = 1) implies that there is also an edge between vertices *j* and *i* (*A*[*j*][*i*] = 1)

Adjacency matrix with a digraph



Digraph

	[0]	[1]	[2]	[3]	[4]
[0]	0	0	0	0	0
[1]	0	0	0	1	1
[2]	0	1	0	1	0
[3]	1	1	0	0	0
[4]	0	1	0	1	0

Equivalent adjacency matrix