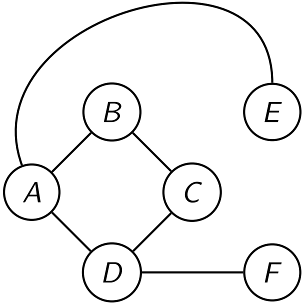
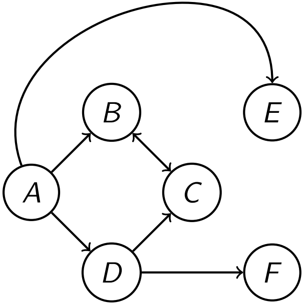
Lecture 1. Introduction to Graph Theory

What is a graph?

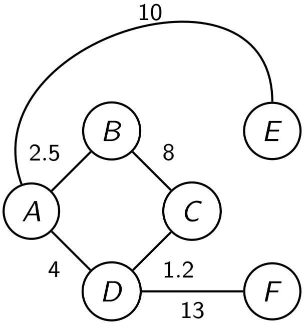
A graph is a collection of vertices (or nodes) which are connected by edges.



(Undirected) Graph



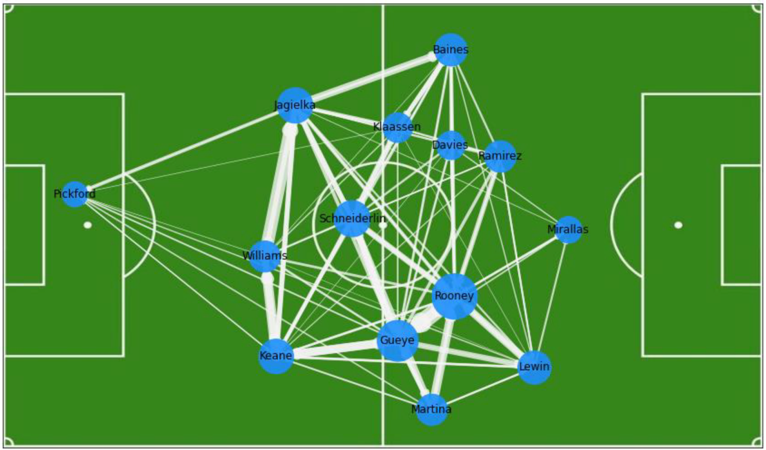
Directed Graph



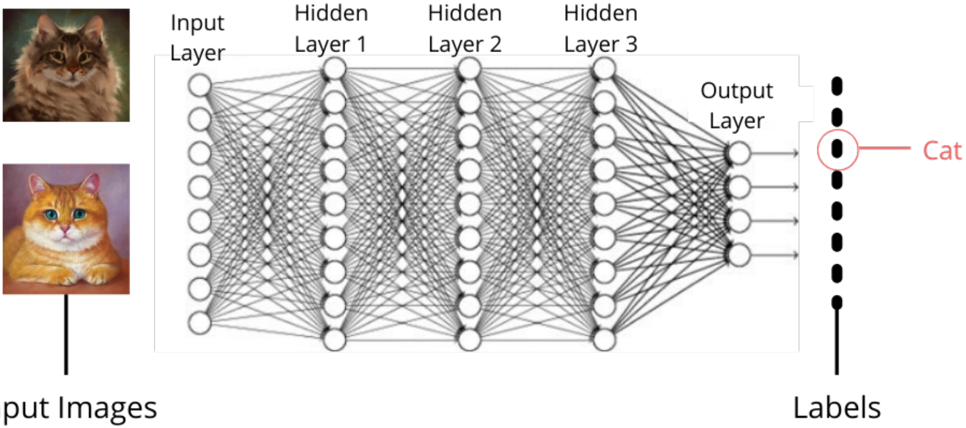
Weighted Graph

The vertices and edges of a graph might represent any number of different things, depending on the application

Graph Model Example 1: Football Passing Networks



Graph Model Example 2: Neural Networks



Graph Model Example 3: Precedence Graphs

We can use a directed graph called a precedence graph to represent which statements must have already been executed before we execute each statement.

- Vertices represent statements in a computer program
- There is a directed edge from a vertex to a second vertex if the second vertex cannot be executed before the first

Example: This precedence graph shows which statements must already have been executed before we can execute each of the six statements in the program.

S_1 $a := 0$

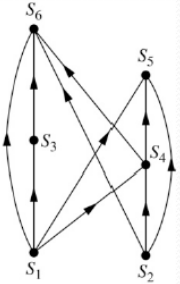
S_2 $b := 1$

S_3 $c := a + 1$

S_4 $d := b + a$

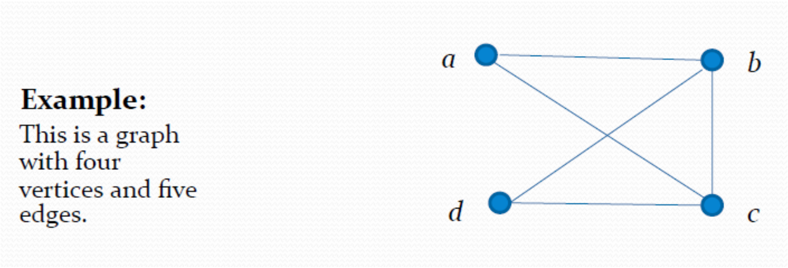
S_5 $e := d + 1$

S_6 $e := c + d$



Terminologies

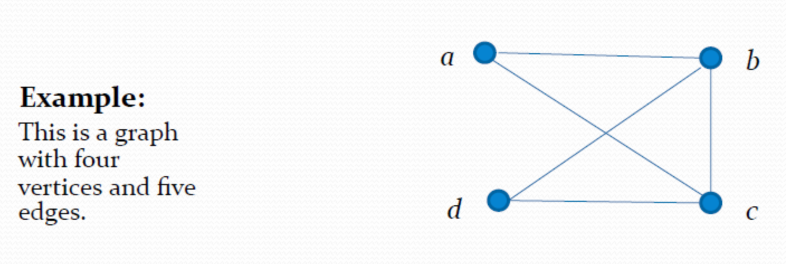
Definition 1.1 A graph $G = (V, E)$ consists of a nonempty set V of **vertices** (or **nodes**) and a set E of **edges**.



where $V = \{a, b, c, d\}$ and $E = \{(a, b), (a, c), (b, d), (b, c), (d, c)\}$.

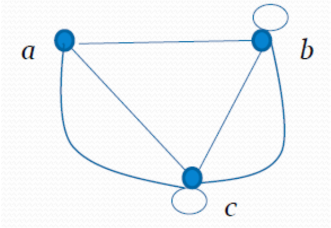
Neighbors: vertices u and v are neighbors (or called adjacent) if an edge (u, v) connects them.

Example 1.2. b and c are neighbors.



Example 1.3. We may ask for all neighbors of the node a: neighbors(a) $= \{b, c\}$

An edge that connects a vertex to itself is called a loop.

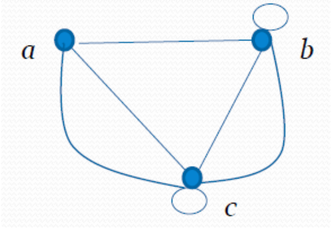


The degree of a vertex is the number of edges connected to the vertex [1], denoted by deg(u). For example, in the figure above

- $\deg(a) =$
- $\deg(b) =$
- $\deg(c) =$

[1] Equivalently, the number of neighbors of the vertex.

An edge that connects a vertex to itself is called a loop.



The degree of a vertex is the number of edges connected to the vertex [1], denoted by deg(u). For example, in the figure above

- $\deg(a) =$
- $\deg(b) =$
- $\deg(c) =$

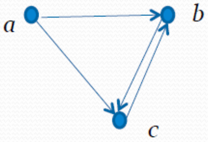
Notice that a loop at a vertex contributes two to the degree of that vertex.

[1] Equivalently, the number of neighbors of the vertex.

Definition 1.4 An directed graph $G = (V, E)$ consists of a nonempty set V of vertices and a set E of directed edges.

Example:

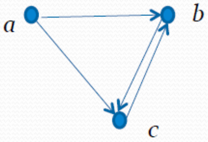
This is a directed graph with three vertices and four edges.



Definition 1.4 An directed graph $G = (V, E)$ consists of a nonempty set V of vertices and a set E of directed edges.

Example:

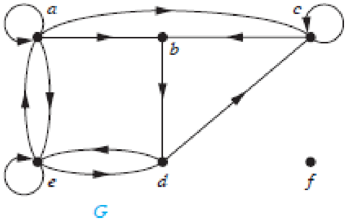
This is a directed graph with three vertices and four edges.



In a directed graph,

- Indegree $=$ the number of edges coming in to the vertex v ($\deg^{-}(v)$)
- Outdegree $=$ the number of edges going out of the vertex v ($\deg^{+}(v)$)

Example 1.5. Find the in-degree and out-degree of each vertex in the graph G with directed edges shown in



A path in a graph G is a sequence of vertices v_i and edges e_i such that the edge e_i connects vertices v_{i-1} and v_i:

$$v_0, e_1, v_1, e_2, v_2, \ldots, v_{n-1}, e_n, v_n$$

Equivalently, assuming that two consecutive vertices are connected by an edge, we may write it without e_i's

$$v_0, v_1, v_2, \ldots, v_{n-1}, v_n$$

A path in a graph G is a sequence of vertices v_i and edges e_i such that the edge e_i connects vertices v_{i-1} and v_i:

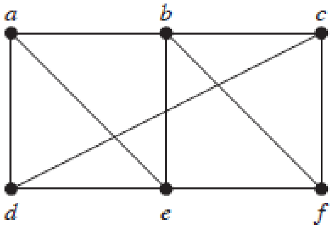
$$v_0, e_1, v_1, e_2, v_2, \ldots, v_{n-1}, e_n, v_n$$

Equivalently, assuming that two consecutive vertices are connected by an edge, we may write it without e_i's

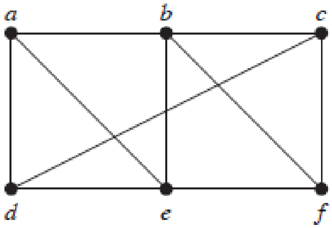
$$v_0, v_1, v_2, \ldots, v_{n-1}, v_n$$

- The length of a path is the number of edges in the path.
- A circuit (or cycle) is a path that starts and ends on the same vertex ($v_0 = v_n$).
- A path or circuit is simple if it does not contain the same edge more than once.

Example. 1.6. In the graph

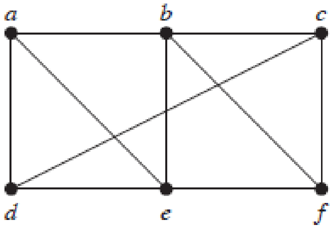


Example. 1.6. In the graph



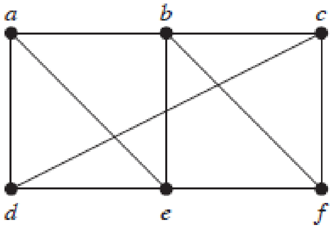
- a, d, c, f, e is a simple path of length 4.

Example. 1.6. In the graph



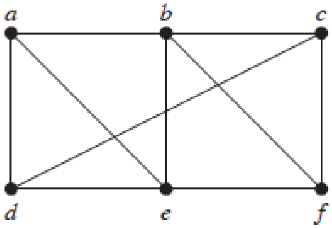
- a, d, c, f, e is a simple path of length 4.
- d, e, c, a is not a path because e is not connected to c.

Example. 1.6. In the graph



- a, d, c, f, e is a simple path of length 4.
- d, e, c, a is not a path because e is not connected to c.
- b, c, f, e, b is a circuit of length 4.

Example. 1.6. In the graph



- a, d, c, f, e is a simple path of length 4.
- d, e, c, a is not a path because e is not connected to c.
- b, c, f, e, b is a circuit of length 4.
- a, b, e, d, a, b is a path of length 5, but not a simple path.

Example 1.7

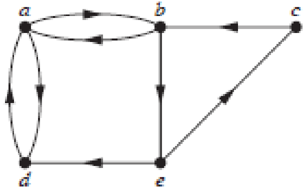
2. Does each of these lists of vertices form a path in the following graph? Which paths are simple? Which are circuits? What are the lengths of those that are paths?

a) a, b, e, c, b

b) a, d, a, d, a

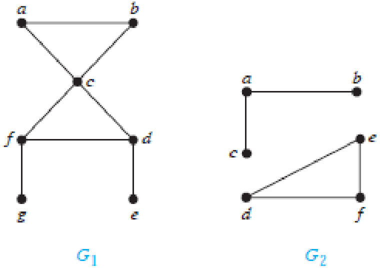
c) a, d, b, e, a

d) a, b, e, c, b, d, a



- An **undirected** graph is connected if there is a path connecting any two vertices.
- A **directed** graph is connected if the underlying undirected graph is.

- An **undirected** graph is connected if there is a path connecting any two vertices.
- A **directed** graph is connected if the underlying undirected graph is.
- **Example 1.8.** G_1 is connected because there is a path between any pair of its vertices, as can be easily seen. However G_2 is not connected because there is no path between vertices a and f, for example.



- Connected component: A subset of vertices $V_i \subset V$ that is connected. For example, in the above graph, $V_1 = \{a, b, c\}$ and $V_2 = \{d, e, f\}$ are two connected components of G_2.

Question 1.9. Is the graph below connected or not?