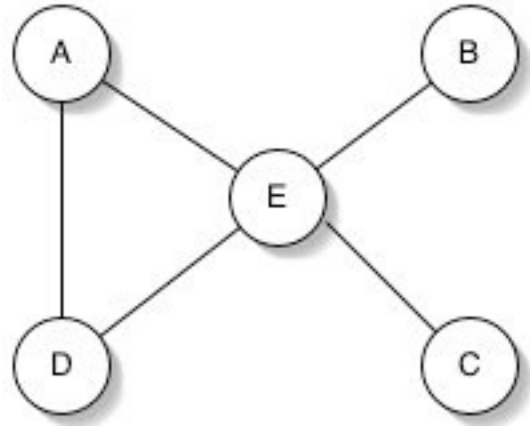
A **graph** is a set of **vertices** and **edges**.

The following graph:



contains the vertices **{A, B, C, D, E}** and the edges
{AE, DE, BE, CE, AD}. These edges are considered **undirected**.

Some terminology:

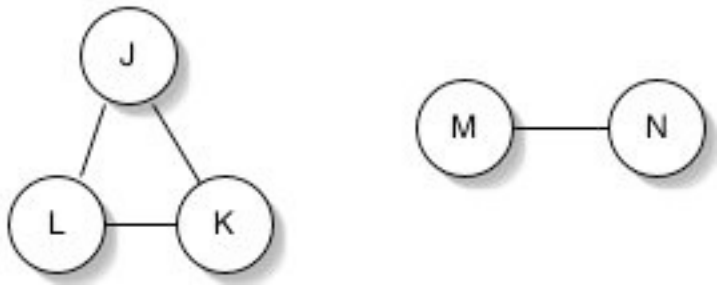
- Graphs can contain **paths** - **B -> E -> A -> D**

- A graph may contain a **cycle** (a path which starts and ends at the same node.) i.e. **A -> E -> D -> A**.

- An **acyclic** graph contains no cycles.

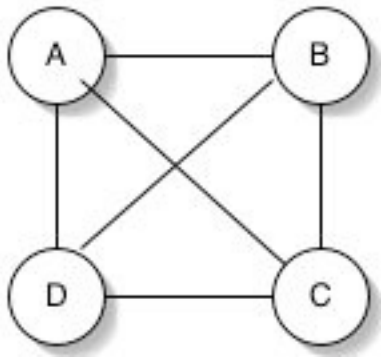
- A graph is **connected** if there is a path from every node to every other node in the graph.

- A **disconnected** graph is comprised of connected components:



G = {J, K, L, M, N}

- Graphs may be **complete** (where there is an edge between every node in the graph)



Graphs may be considered either:

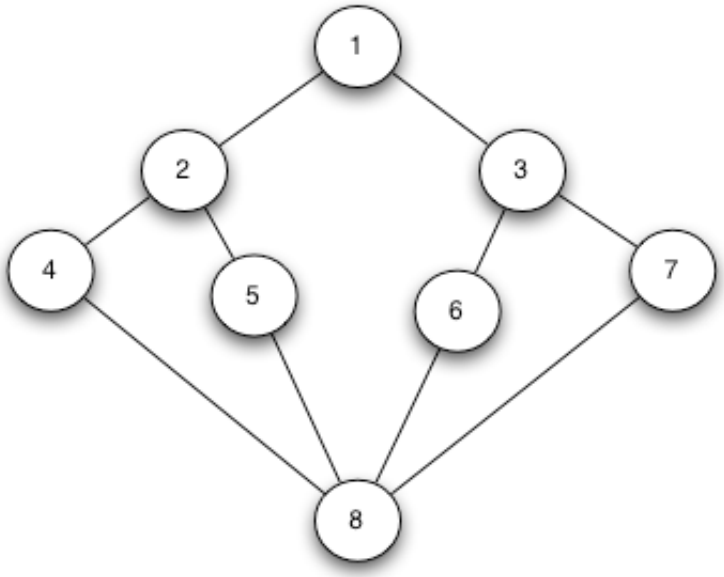
Sparse - with relatively few edges

Dense - with relatively few edges missing.

Graphs may also be **directed** where the edge goes only one way:

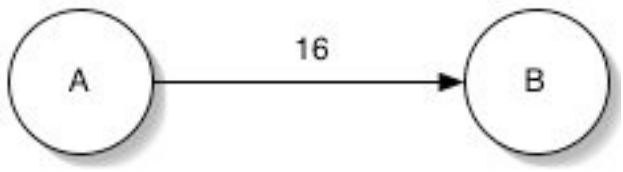
Bipartite Graphs

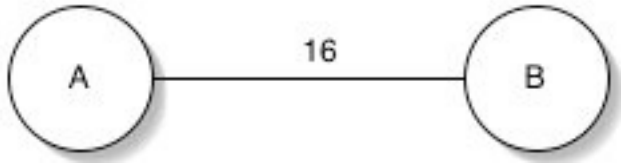
A **bipartite graph** is an undirected graph in which the vertices can be partitioned into two subsets V1 and V2 where all edges go between the two sets. For the graph below



a bipartite partitioning is **V1 = {1,4,5,6,7} V2 = {2, 3, 8}**

Graphs may also be **weighted**:





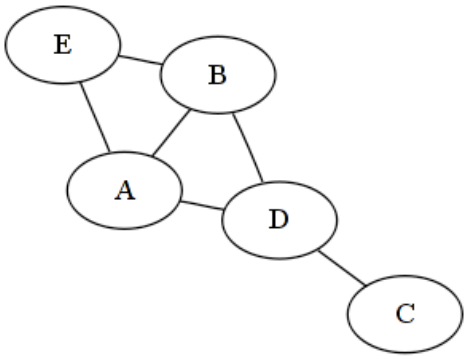
When graphs are weighted, we will use the following notation for an edge: **{ (A, B, 16) }**. Note that whether the edge is directed or not depends on how it is **interpreted**. The edge

{ (A, B, 16) } may be considered either directed or undirected.

Representation of Graphs:

There are two general approaches for representing a graph, an **adjacency list** and an **adjacency matrix**.

For the following undirected, unweighted graph:



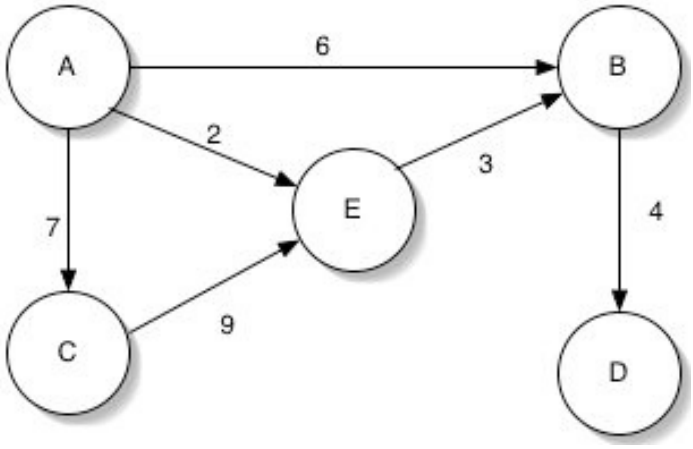
Adjacency List

A: B, D, E
B: A, D, E
C: D
D: A, B, C
E: A, B

Adjacency Matrix

	A	B	C	D	E
A		1		1	1
B	1			1	1
C				1	
D	1	1	1		
E	1	1			

Consider the following weighted, directed graph:



Adjacency Matrix

	A	B	C	D	E
A		6	7		2
B				4	
C					9
D					
E		3			

Adjacency List

A = (B, 6), (C, 7), (E, 2)

B = (D, 4)

C = (E, 9)

E = (B, 3)