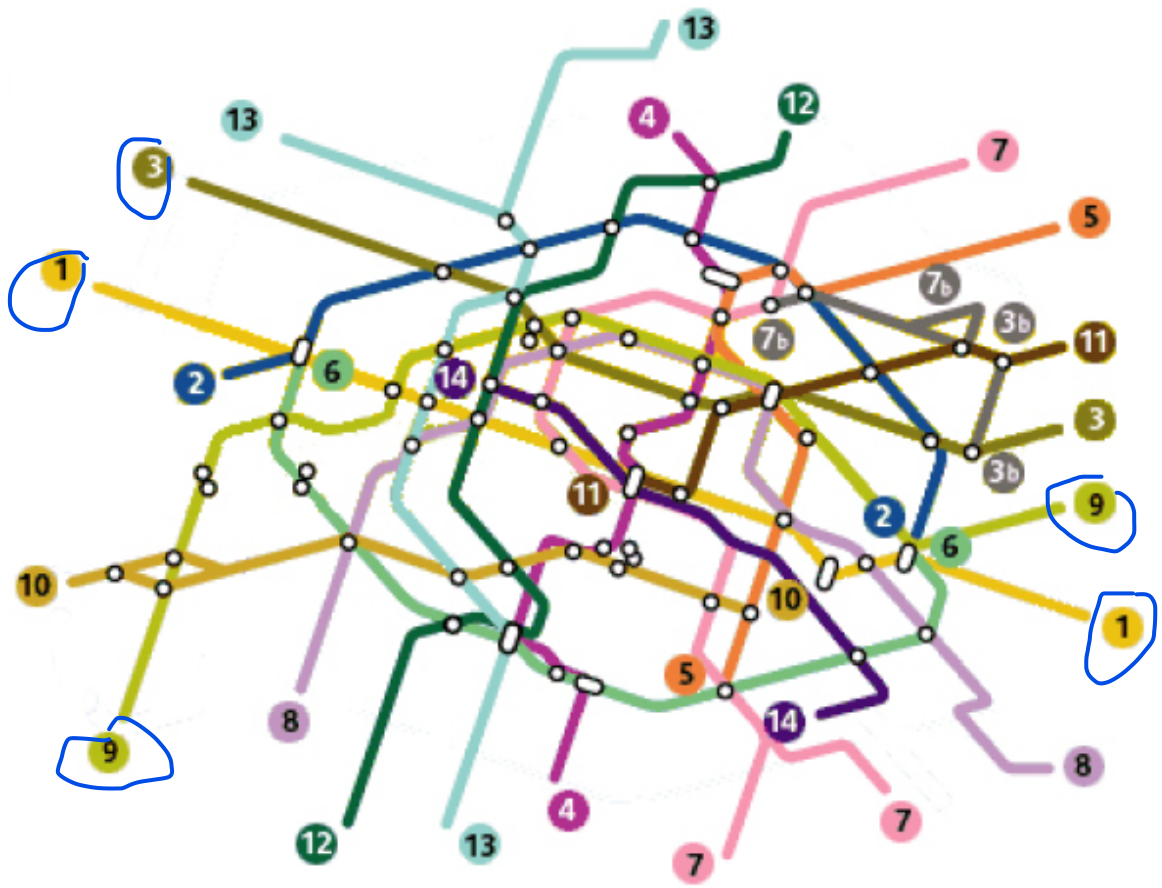
WHAT ARE GRAPHS?

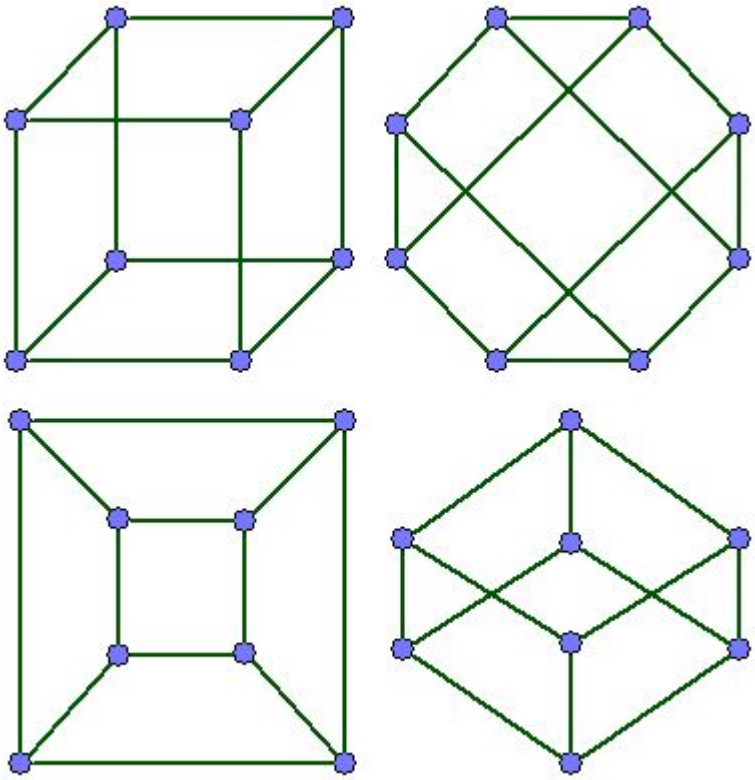
Graphs are structures made up of a certain number of **vertices(or nodes)** connected to each other by **edges or arcs** depending on the type of graph. We can define a **Graph** `G` by example `G = (V,E)` where **V** is the set of vertices of `G` and `E` is the set of edges.

- `G` is the graph
- `V` is the list of vertices / nodes
- `E` is the list of edges / arcs

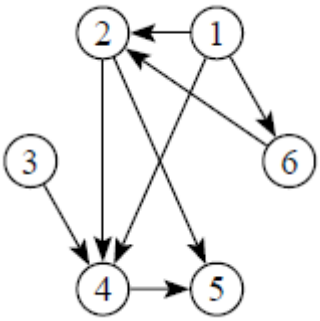
`G(V,E)` is `G` the graph `V={V1,V2,...,Vn}` the list of nodes and `E =` {E1,E2,...,En}` the list of edges that correspond to a **pair** representing and **link** between two **nodes**.

Example1:





Example2:



Example3:

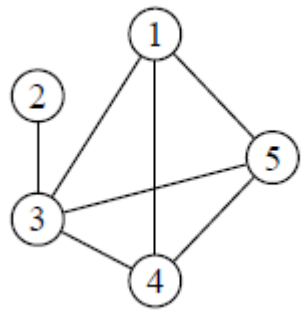
We can see that we have two types of Graph:

- **undirected graph** where edges have no direction. which means that connection between two nodes are bidirectional you can travel the two nodes in both end points. if a edge `A` and `B` are connected you travel from `A` to `B` and vice versa.
- **directed graph** where edges have a direction. meaning that connection between two nodes is **one-way**. if an edge from `A` to `B` exists it does not necessarily means that there's an edge from `B` to `A`.

Degree of a Node

the "degree of a node" refers to *the number of edges that are connected to that node*.if the "degree of a node" is high it usually means that the node is complex and also that the node is

important. the degree of a node is usually noted `deg(node) = a_number`



in this graph we can see the the degree of the different node:

`d(1) = 3`

`d(2) = 1`

`d(3) = 4`

`d(4) = 3`

`d(5) = 3`

Clustering in a Graph

basically clustering allows us to know what are the most important group of vertices in a graph. in the earlier graph example we can see that the nodes are grouped around the `node 3`

Representing Graph in Algorithm

when doing algorithm with graph we usually represent it in two form:

- **Matrices** by using Matrix we can represent a graph where the rows and columns represent the vertices and their connection is the intersection between row x column. the size of the Matrix depend on the number of node.
- **dictionary of list** where every entry in the dictionary represent a node follow by an list of the node connected to the current node.