There are two ways in which we represent graphs, these are:

- Adjacency Matrix
- Adjacency List

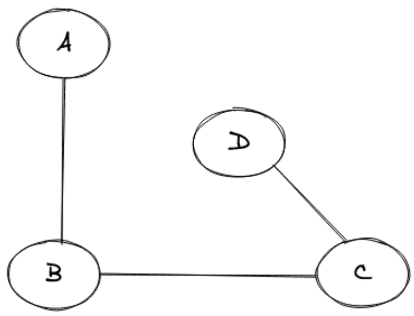
Both these have their advantages and disadvantages.

Adjacency Matrix

Adjacency matrix representation makes use of a matrix (table) where the **first row and first column of the matrix denote the nodes** (vertices) of the graph. The **rest of the cells contains either 0 or 1** (can contain an associated **weight w** if it is a weighted graph).

Each **row X column** intersection points to a cell and the value of that cell will help us in determining that whether the vertex denoted by the row and the vertex denoted by the column are connected or not. If the value of the cell for v1 X v2 is equal to 1, then we can conclude that these two vertices v1 and v2 are connected by an edge, else they aren't connected at all.

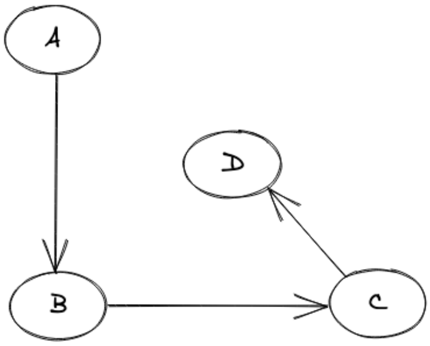
Adjacency matrix of the undirected graph:



-	A	B	C	D
A	0	1	0	0
B	1	0	1	0
C	0	1	0	1
D	0	0	1	0

we can see that the above matrix is symmetric (square matrix).

Adjacency matrix of the directed graph:

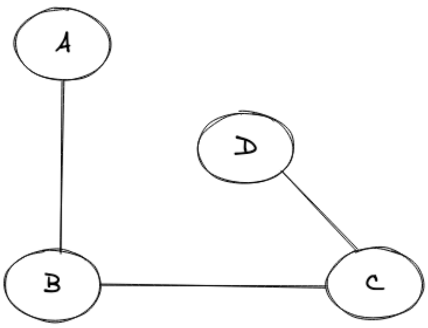


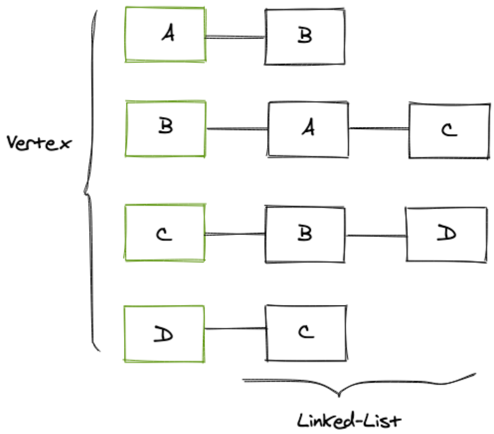
-	A	B	C	D
A	0	1	0	0
B	0	0	1	0
C	0	0	0	1
D	0	0	0	0

Adjacency List

In the adjacency list representation, we have an array of linked-list where the size of the array is the number of the vertex (nodes) present in the graph. Each vertex has its own linked-list that contains the nodes that it is connected to.

Adjacency List of the undirected graph





The first column contains all the vertices we have in the graph above and then each of these vertices contains a linked list that in turn contains the nodes that each vertex is connected to. For a directed graph the only change would be that the linked list will only contain the node on which the incident edge is present.

Adjacency List of the directed graph