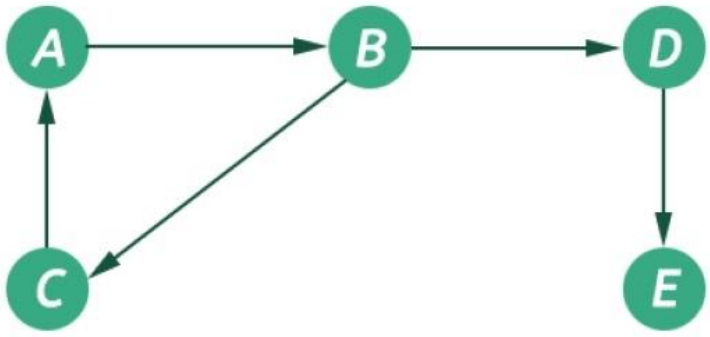
REPRESENTATION OF GRAPHS

GRAPH

- The graph is a non-linear data structures.
- This represents data using nodes, and their relations using edges.
- A graph G has two sections.
  I. The vertices
  II. Edges
- Vertices are represented using set V, and Edges are represented as set E.
- So the graph notation is G (V, E).

EXAMPLE



- In this graph, there are five vertices and five edges.
  V = {A, B, C, D, E}
  E = {<A,B>,<B,D>,<B,D>,<C,A>,<D,E>}
- The edges are directed. As an example, if we choose the edge connecting vertices B and D, the source vertex is B and destination is D. So we can move B to D but not move from D to B.

REPRESENTATION OF GRAPHS

The two most common ways of representing a graph is as follows:

1. **Adjacency matrix representation**
2. **Adjacency List representation**

Adjacency matrix

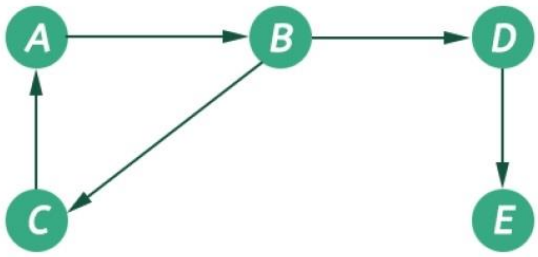
- An **adjacency matrix** can be thought of as a table with rows and columns.
- The row labels and column labels represent the nodes of a graph.

- An adjacency matrix is a square matrix where the number of rows, columns and nodes are the same.
- Each cell of the matrix represents an edge or the relationship between two given nodes.

- An adjacency matrix is a **VxV** binary matrix **A**.
- Element $A_{i, j}$ is 1 if there is an edge from vertex i to vertex j else $A_{i, j}$ is 0.

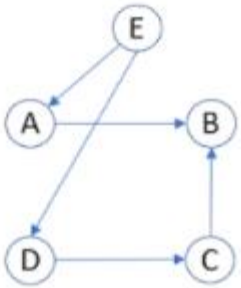
- *Note*: A binary matrix is a matrix in which the cells can have only one of two possible values - either a 0 or 1.

- The adjacency matrix can also be modified for the weighted graph in which instead of storing 0 or 1 in $A_{i, j}$, the weight or cost of the edge will be stored.
- In an undirected graph, if $A_{i, j} = 1$, then $A_{j,i} = 1$. In a directed graph, if $A_{i, j} = 1$, then $A_{j,i}$ may or may not be 1.



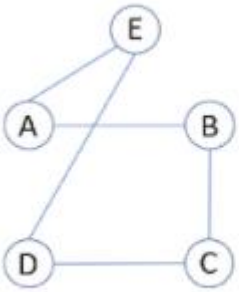
The adjacency matrix of the above graph is:

	A	B	C	D	E
A	0	1	0	0	0
B	0	0	1	1	0
C	1	0	0	0	0
D	0	0	0	0	1
E	0	0	0	0	0



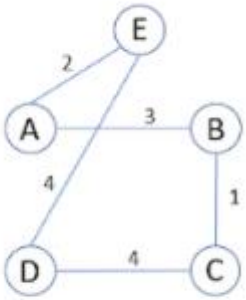
	A	B	C	D	E
A	0	1	0	0	0
B	0	0	0	0	0
C	0	1	0	0	0
D	0	0	1	0	0
E	1	0	0	1	0

Fig 3: Adjacency Matrix for a directed graph



	A	B	C	D	E
A	0	1	0	0	1
B	1	0	1	0	0
C	0	1	0	1	0
D	0	0	1	0	1
E	1	0	0	1	0

Fig 4: Adjacency Matrix for an undirected graph

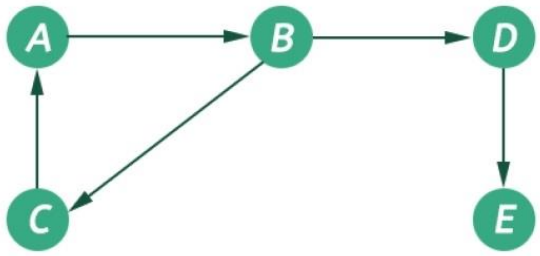


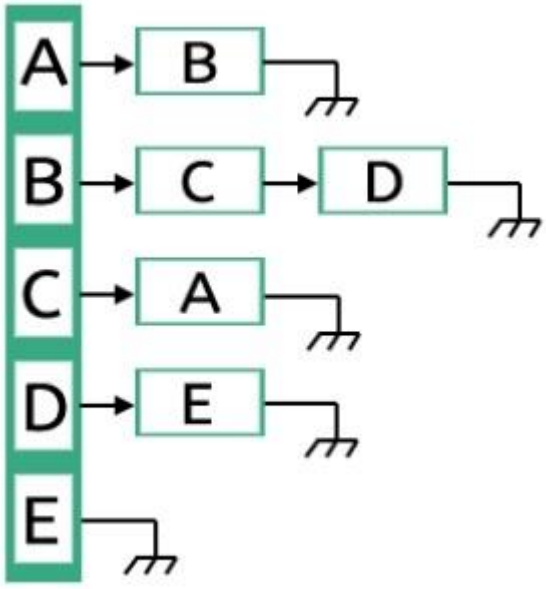
	A	B	C	D	E
A	0	3	0	0	2
B	3	0	1	0	0
C	0	1	0	4	0
D	0	0	4	0	4
E	2	0	0	4	0

Fig 5: Adjacency Matrix for a weighted graph

Adjacency List Representation

- This representation is based on Linked Lists.
- An adjacency list is an array A of separate lists.
- Each element of the array **Ai** is a list, which contains all the vertices that are adjacent to vertex i.
- At the end of list, each node is connected with the null values to tell that it is the end node of that list.





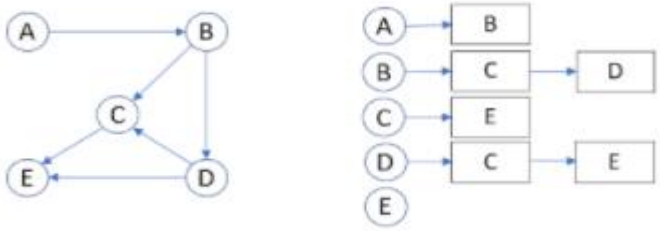


Fig 6: Adjacency list for a directed graph

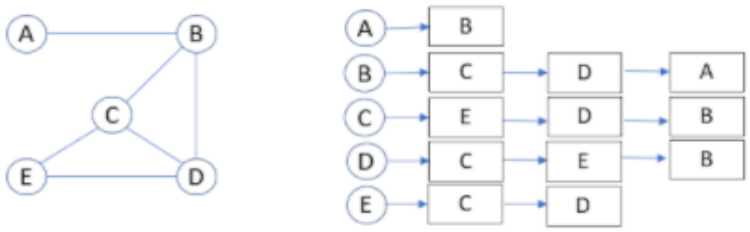


Fig 7: Adjacency list for an undirected graph