5.1:Graph Theory

Graph Representation

Graph Representations

- In graph theory, a graph representation is a technique to store graph into the memory of computer.
- To represent a graph, we just need the set of vertices, and for each vertex the neighbors of the vertex (vertices which is directly connected to it by an edge).
- There are different ways to represent a graph:
  - Adjacency Matrix
  - Incidence Matrix
  - Adjacency List

Adjacency Matrix

- Adjacency matrix is a sequential representation.
- It is used to represent which nodes are adjacent to each other. i.e. is there any edge connecting nodes to a graph.
- In this representation, we have to construct a nXn matrix A.
- The adjacency matrix of G with respect to given ordered list of vertices is a n*n matrix denoted by $A(G)=(a_{ij})_{n*n}$ such that

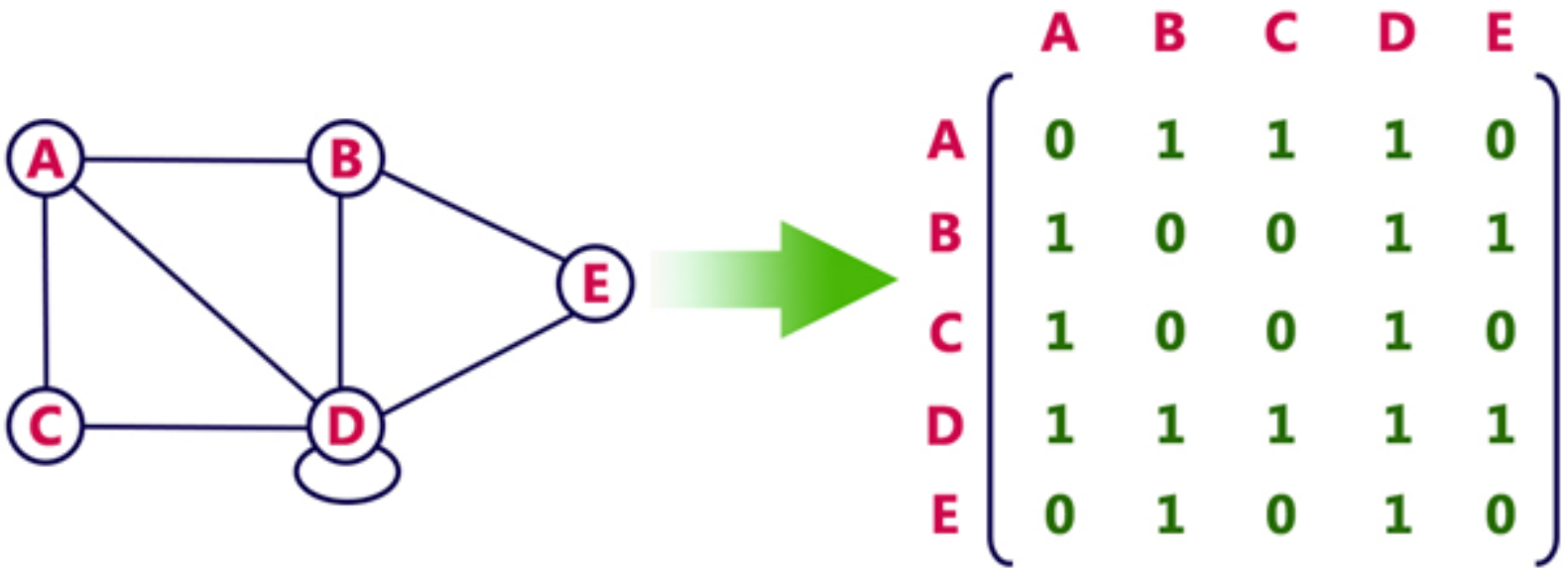
  a_{ij} = 0 if there is no edge between the vertices v_i

  1 if there is edge between the vertices v_i

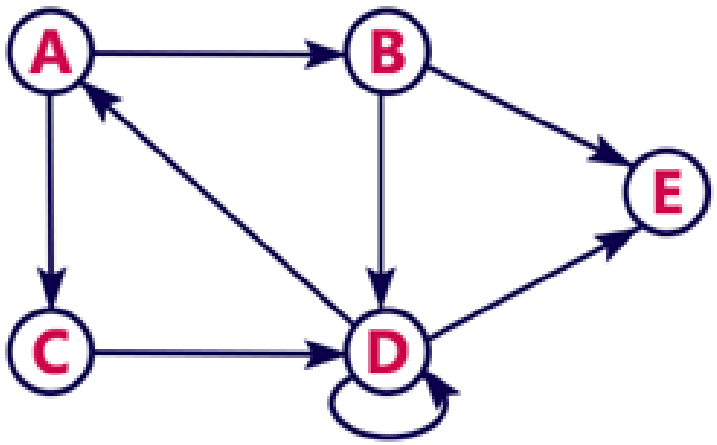
  K if there are K (K>=2) edges between the vertices v_i

Example

- Consider the following **undirected graph representation**:



Example: For Directed graph



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

- **Pros:** Representation is easier to implement and follow.
- **Cons:** It takes a lot of space and time to visit all the neighbors of a vertex, we have to traverse all the vertices in the graph, which takes quite some time.

Incidence Matrix

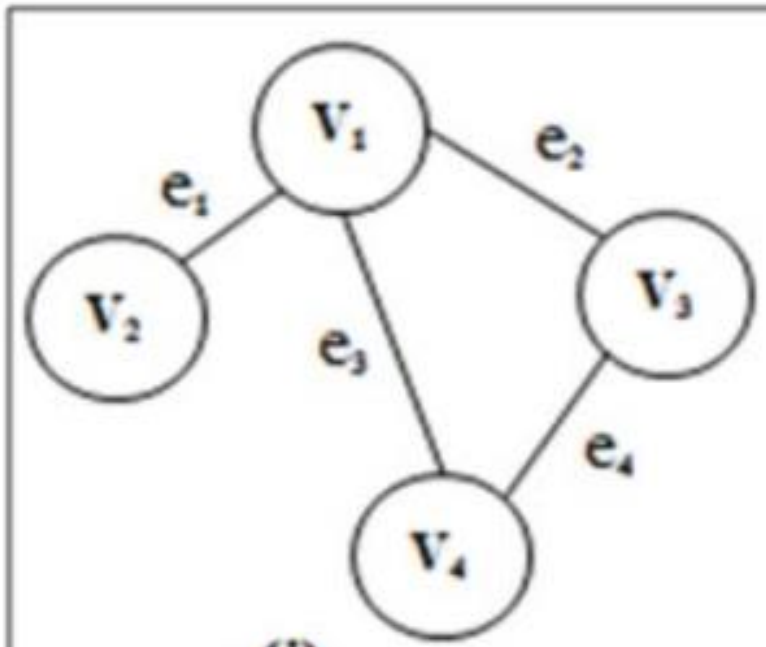
- In **Incidence matrix representation**, graph can be represented using a matrix of size:
  Total number of vertices by total number of edges.
- It means if a graph has 4 vertices and 6 edges, then it can be represented using a matrix of 4X6 class. In this matrix, columns represent edges and rows represent vertices.
- Let G be a graph with vertices $v_1, v_2, \ldots\ldots, v_m$ and edes $e_1, e_2, \ldots\ldots, e_n$. The incidence matrix I(G) of graph G is a m*n matrix with $I(G)= (m_{ij})_{m*n}$ where

m_{ij} = 1 if ej is incident with vi

0 if ej is not incident with vi

2 if vi is has loop

Example

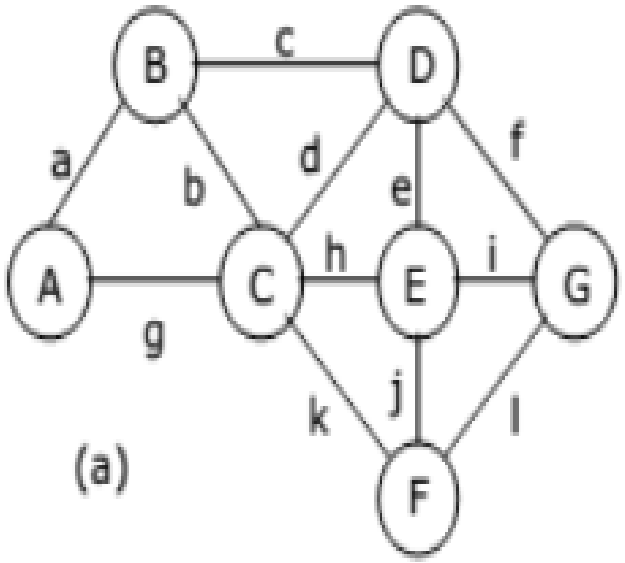


(i)

	e_1	e_2	e_3	e_4
v_1	1	1	1	0
v_2	1	0	0	0
v_3	0	1	0	1
v_4	0	0	1	1

(ii)

Example



(a)

(b)

	a	b	c	d	e	f	g	h	i	j	k	l
A	1	0	0	0	0	0	1	0	0	0	0	0
B	1	1	1	0	0	0	0	0	0	0	0	0
C	0	1	0	1	0	0	1	1	0	0	1	0
D	0	0	1	1	1	1	0	0	0	0	0	0
E	0	0	0	0	1	0	0	1	1	1	0	0
F	0	0	0	0	0	0	0	0	0	1	1	1
G	0	0	0	0	0	1	0	0	1	0	0	1

Incidence Matrix: For directed graph

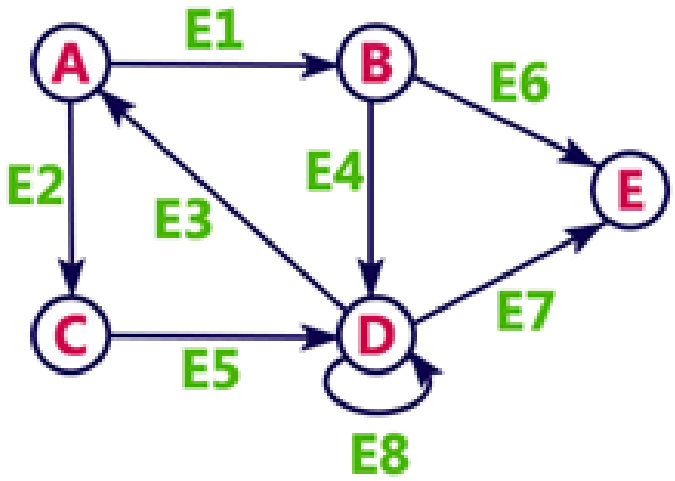
- This matrix is filled with either **0 or 1** or -1. Where,

0 is used to represent row edge which is not connected to column vertex.

1 is used to represent row edge which is connected as outgoing edge to column vertex.

-1 is used to represent row edge which is connected as incoming edge to column vertex.

Example



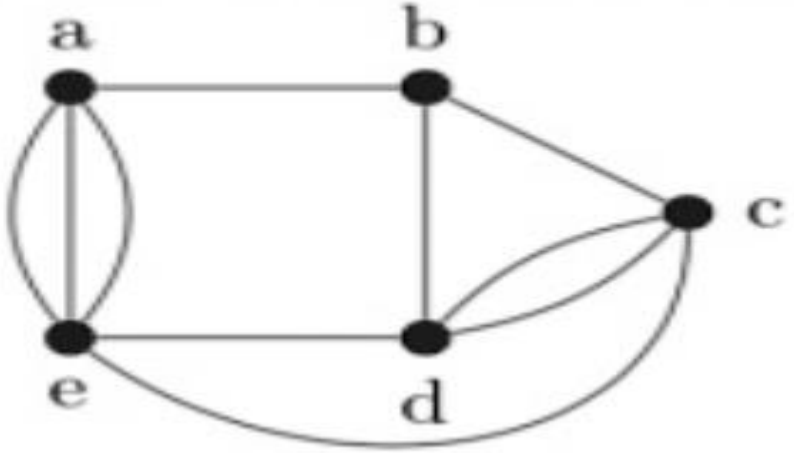
	E1	E2	E3	E4	E5	E6	E7	E8
A	1	1	-1	0	0	0	0	0
B	-1	0	0	1	0	1	0	0
C	0	-1	0	0	1	0	0	0
D	0	0	1	-1	-1	0	1	1
E	0	0	0	0	0	-1	-1	0

Questions

Draw an undirected graph from the adjacency matrix.

$$\begin{bmatrix} 0 & 1 & 3 & 0 & 4 \\ 1 & 2 & 1 & 3 & 0 \\ 3 & 1 & 1 & 0 & 1 \\ 0 & 3 & 0 & 0 & 2 \\ 4 & 0 & 1 & 2 & 3 \end{bmatrix}$$

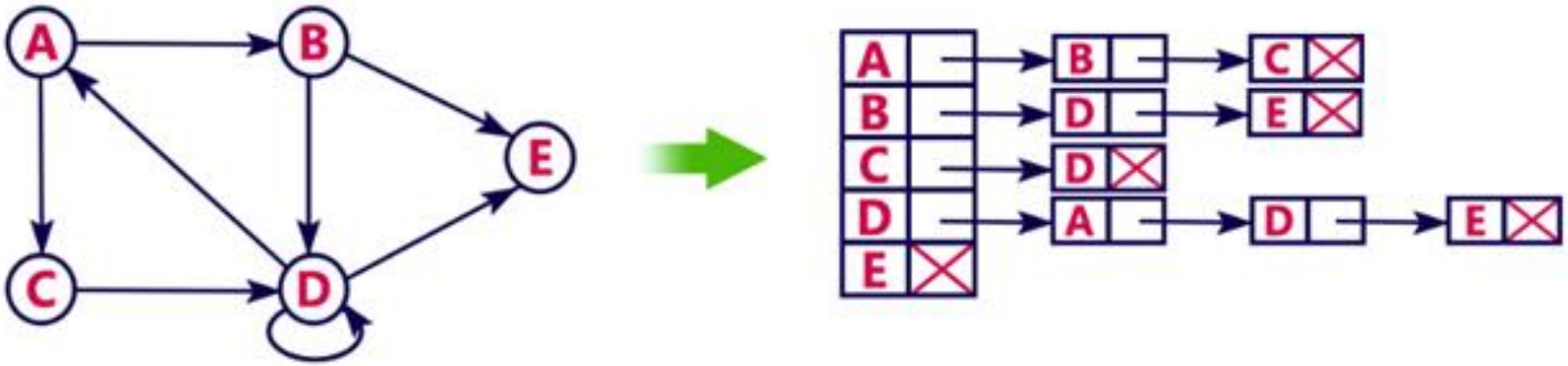
Write an incidence matrix for the graph shown.



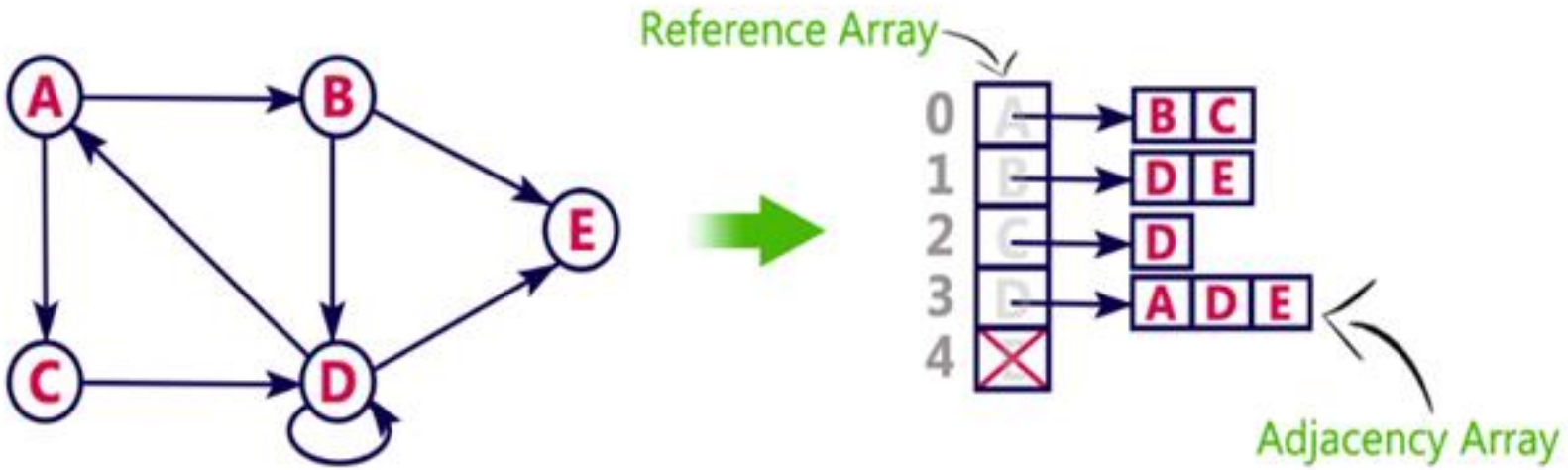
Adjacency List

- Adjacency list is a linked representation.
- In this representation, for each vertex in the graph, we maintain the list of its neighbors.
- It means, every vertex of the graph contains list of its adjacent vertices.
- We have an array of vertices which is indexed by the vertex number and for each vertex v, the corresponding array element points to a singly linked list of neighbors of v.

Example

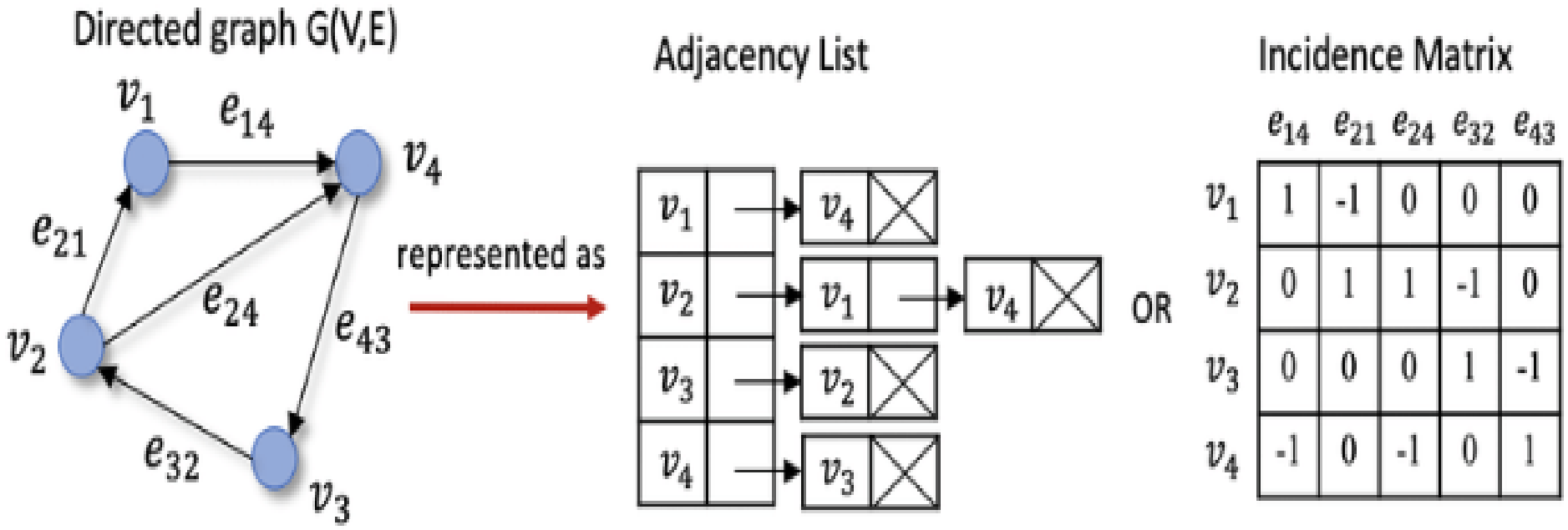


- We can also implement this representation using array as follows:

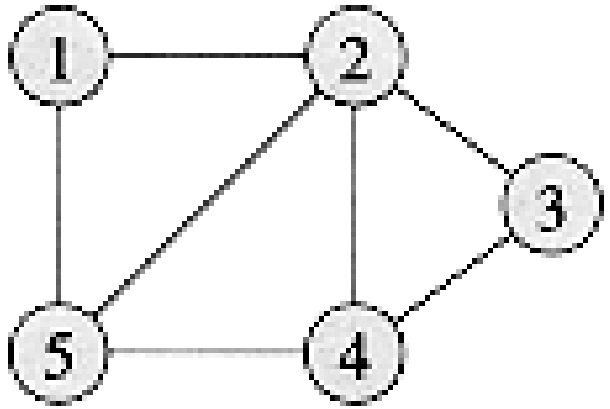


- **Pros:**
  - Adjacency list saves lot of space.
  - We can easily insert or delete as we use linked list.
  - Such kind of representation is easy to follow and clearly shows the adjacent nodes of node.
- **Cons:**
  - slower

Example

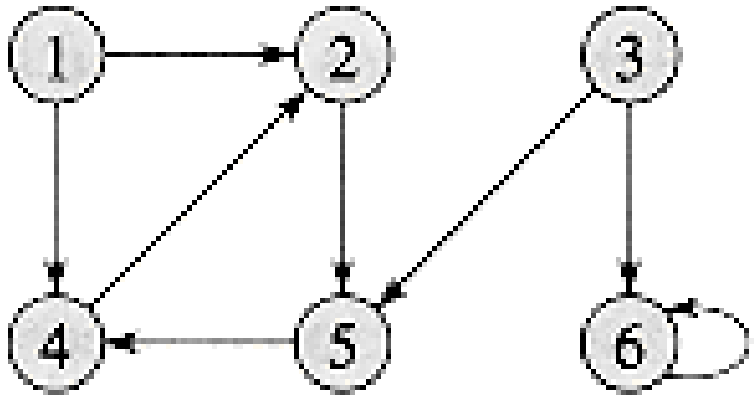

Directed graph G(V,E)
v_1
e_{14}
v_4
e_{21}
e_{24}
e_{43}
v_2
e_{32}
v_3
represented as
Adjacency List
v_1
v_4
v_2
v_1
v_4
v_3
v_2
v_4
v_3
OR
Incidence Matrix
e_{14} e_{21} e_{24} e_{32} e_{43}
v_1 1 -1 0 0 0
v_2 0 1 1 -1 0
v_3 0 0 0 1 -1
v_4 -1 0 -1 0 1

Question: Write adjacency matrix, incidence matrix and Adjacency list



(a)

Question: Write adjacency matrix, incidence matrix and Adjacency list



(a)