CSE221 ALGORITHMS

Topic: Graph (Directed, Undirected, Unweighted), Adjacency Matrix, Adjacency List, Tree

Prepared by:

Saad Bin Sohan

BRAC University

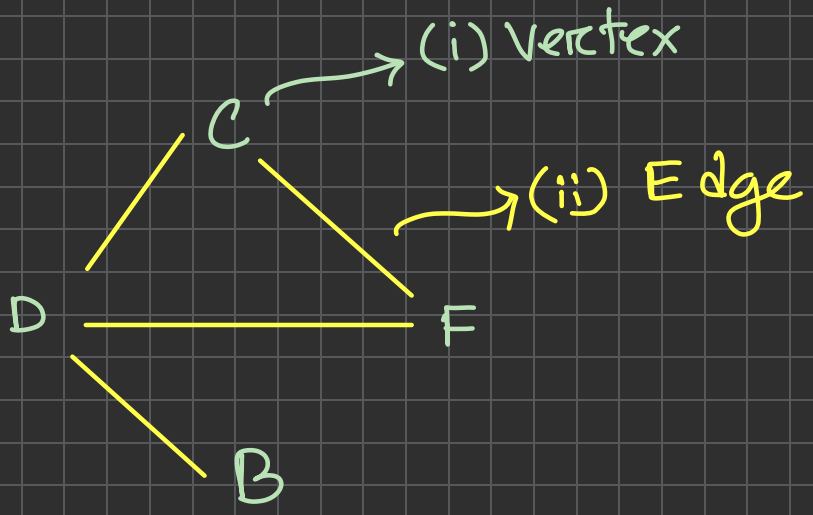
Email: sohan.academics@gmail.com

GitHub: https://github.com/saad-bin-sohan

Topic: Graph

Graph

Basic two parts of a graph data structure:



Imagine 'vertex' as cities and 'Edge' as roads connecting them or linking them.

Types of Graph

Primarily two types:

(i) undirected

(ii) Directed

[redacted]

Graph can also be categorized as

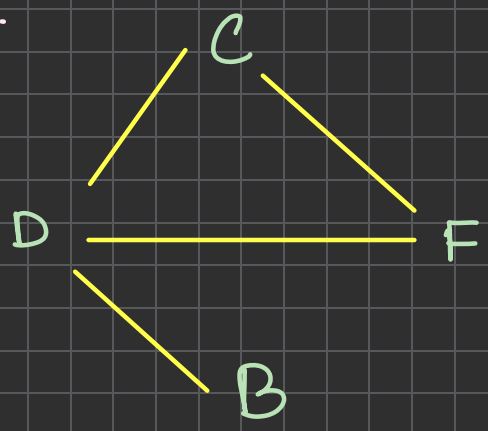
(a) weighted

(b) unweighted

Undirected Graph

No particular direction is given from one vertex (city) to another vertex. Both direction ~~bnt~~ possible.

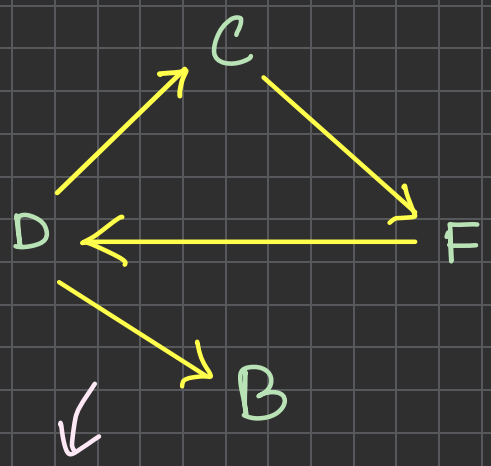
যেমন:



Directed Graph

→ Directions from one vertex to adjacent vertices are given. And it allows only one way traverse (চলাচল).

যেমন:



it can also be called unweighted graph

[redacted]

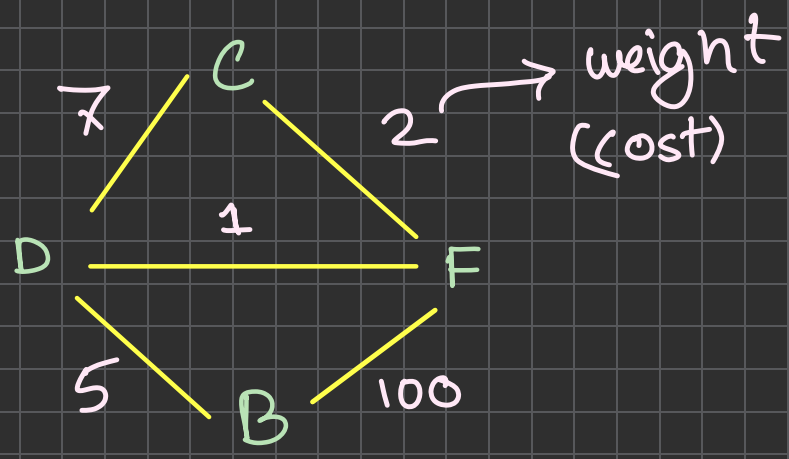
NB: → Both way direction দেওয়া থাকে না।

→ Direction দেওয়া না থাকলে means both way চলাচল

[redacted]

Unweighted Graph

→ Weight বলা নাই, e.g. এক city (vertex) থেকে আরেক vertex এ যেতে cost (weight) কত। যেমন:



→ either সবগুলোতে weight বলা থাকবে or কোনোটাতে weight বলা থাকবে না।

[redacted]

Important properties of Graph:

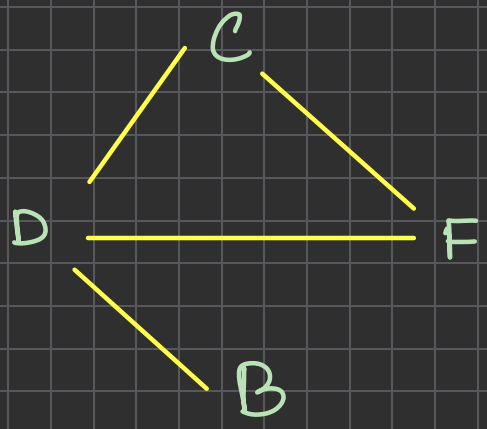
Properties of Undirected Graph:

(i) Degree: number of edges connected to the vertex.

যেমন:

[redacted]

Vertex	Degree
B	1
C	2
D	3
F	2



sum of degrees = 8

notice something?

Degree sum of all vertices,

= 2 × number of edges

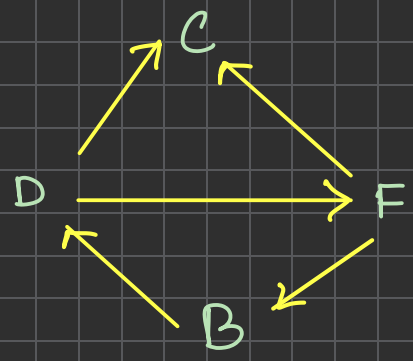
<u>reason:</u> each edge connects exactly two vertices. thats why increase two times

Properties of Directed Graph:

(i) Indegree: number of incoming edges

(ii) Outdegree: number of outgoing edges

Vertex	Indegree	Outdegree
B	1	1
C	2	0
D	1	2
F	1	2
	sum = 5	sum = 5



notice something?

$$\Sigma\,(\text{in degree}) = \Sigma\,(\text{out degree})$$

sum of indegree = sum of outdegree

Representation of Graph

Two methods to represent graph:

- → Adjacency Matrix
- → Adjacency List

[redacted]

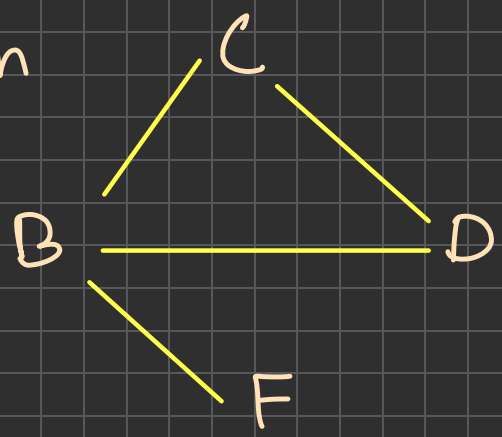
Adjacency Matrix

↳ Table / 2d array

Vertex সংখ্যা n হলে,

matrix size = $n \times n$

→ start with a $n \times n$ array where column title and row title is each vertex one by one



[redacted]

→ then, if there is an edge between a vertex 'x' (row) and vertex 'y' (column) then insert '1' in the mutual box of vertex 'x' and 'y'

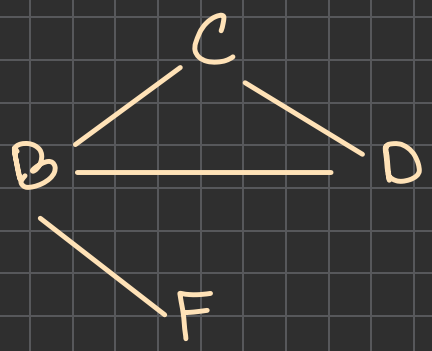
And if there is no edge between the two vertices, then insert '0' (zero) in the box.

0 → no edge

1 → edge exists

And you have a matrix representation of Graph

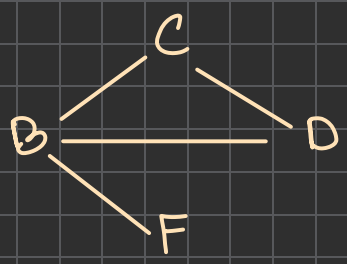
For example,



This graph can be represented as follows,

[redacted]

	B	C	D	F
B	0	1	1	1
C	1	0	1	0
D	1	1	0	0
F	1	0	0	0



Reason for B×B being zero is we can't traverse from B to B in the given example.

And if it was like 

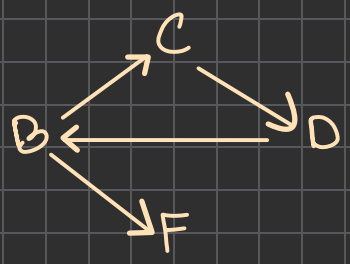
then B×B would be 1.

Directed graphs can also be represented in matrix. But the direction will be:

$$\underbrace{\text{row} \rightarrow \text{column}}_{\text{always}}$$

For example,

	B	C	D	F
B	0	1	0	1
C	0	0	1	0
D	1	0	0	0
F	0	0	0	0



[redacted]

Adjacency List

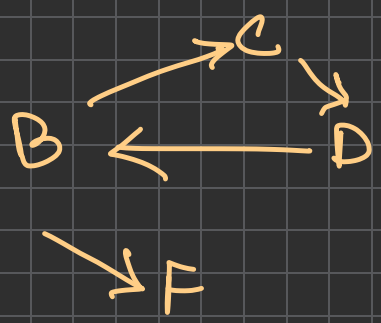
↳ list of lists

→ start with a list with all individual vertices

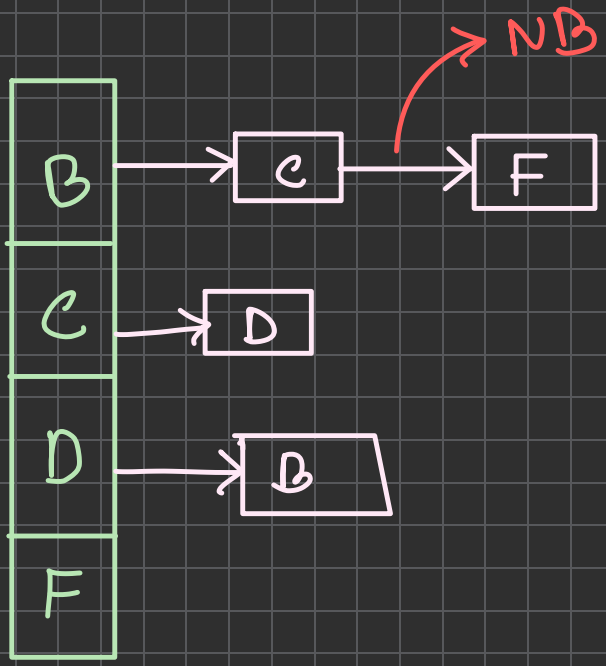
→ then the vertices in that list will be followed by all the vertices that have edges with that vertex from the main list and the

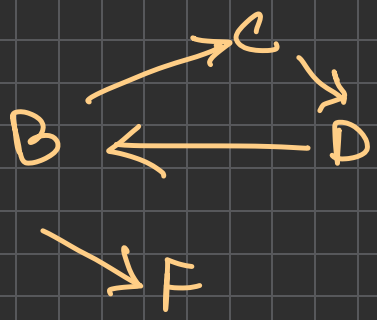
following vertices (with connecting edges) will be saved as list.

For example,



can be represented in Adjacency List as follows,





NB: there C → F doesn't mean we can go from C to F or that C and F have mutual edge. Because in the graph we see they don't. What that

[redacted]

means is we can go from B to both C and F directly. Meaning C and F both have edges from B.

[redacted]

Tree

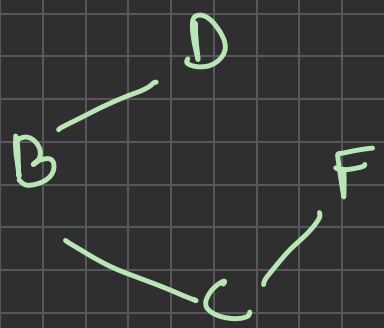
Tree is a graph which maintains 3 conditions.

Conditions:

(i) Connected হতে হবে:

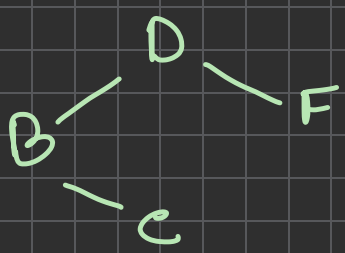
যেকোনো একটা vertex থেকে যাত্রা শুরু করলে somehow অন্য সকল vertex এ যাওয়া যাবে without lifting the pen.

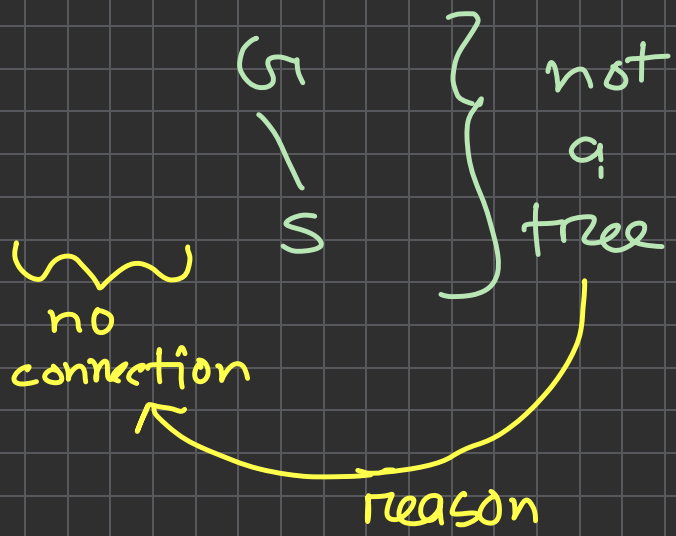
যেমন:



→ its a tree

but



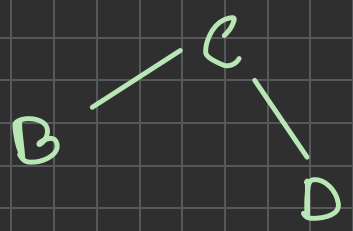


we cant go from D to S without lifting the pen.

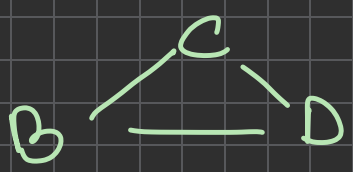
That's still a graph but not a tree.

disconnected graph

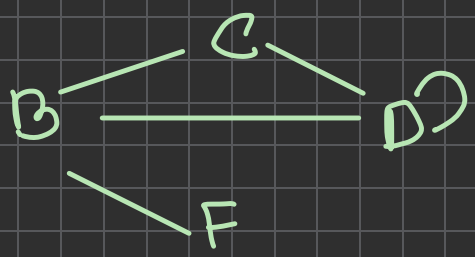
condition (ii) No cycle whatsoever!



→ ✓ tree



→ not a tree



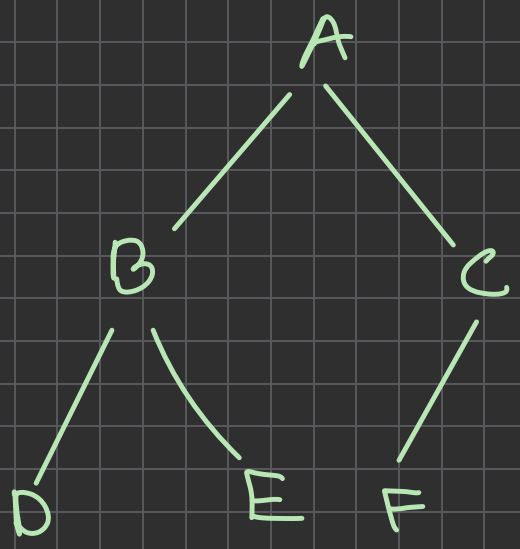
→ not a tree

<u>condition (iii):</u> Graph এ exactly (n-1) সংখ্যক edges থাকা লাগবে where

n = number of vertices.

[redacted]

Tree:



$n = 6$

edges $= 5$