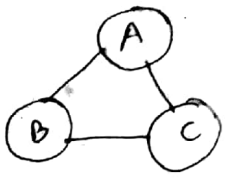
Graph

What is graph?



Vertex

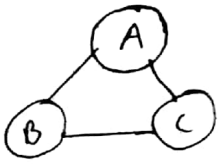
V = Vertices = {A, B, C}

E = Edges = {AB, BC, AC}

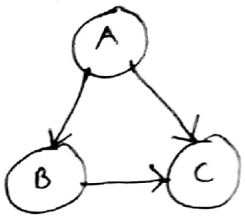
Edge

Graph, G = (V, E)

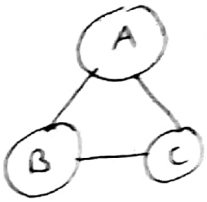
Different types of Graph



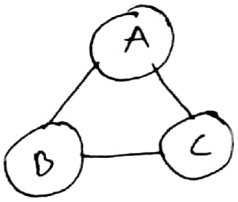
Undirected



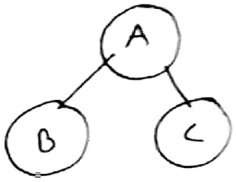
Directed



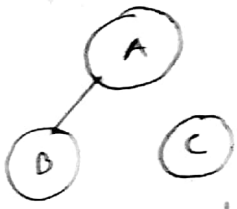
connected



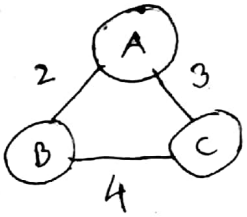
cyclic



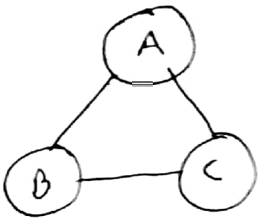
Acyclic



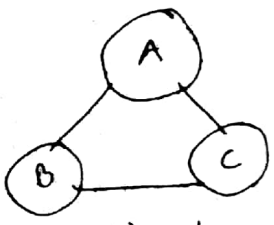
Disconnected



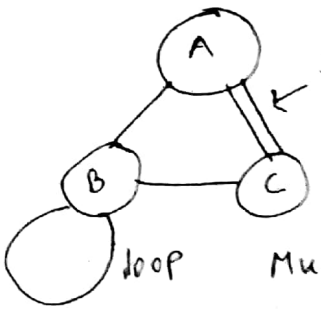
Weighted



Unweighted

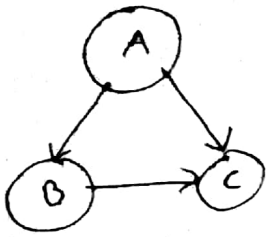


Simple



Adjacency structure of Graph

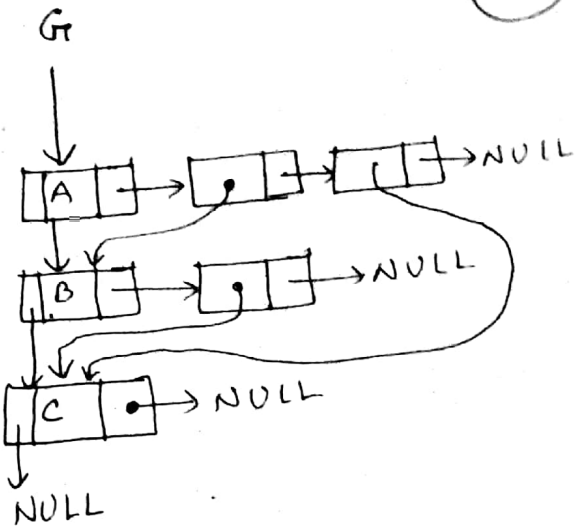
- Array
- Linked list



Graph

	A	B	C
A	0	1	1
B	0	0	1
C	0	0	0

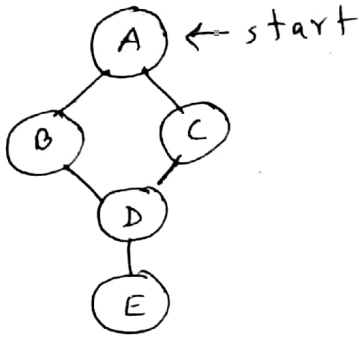
Array



Linked list

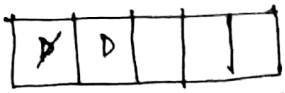
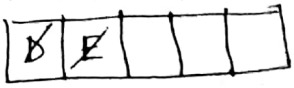
Graph Traversal Techniques

- BFS (Breadth First Search)
- DFS (Depth First Search)



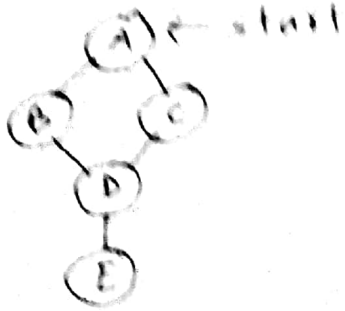
BFS:

Queue 

v ⇐ Queue	Visit v	Adjacent of v is u	Queue ⇐ u
v = 'A'	A	u = {B, C}	Queue 
v = 'B'	B	u = {~~A~~, D}	Queue 
v = 'C'	C	u = {~~A~~, D}	Queue 
v = 'D'	D	u = {~~B~~, ~~C~~, E}	Queue 
v = 'D'	X	X	X
v = 'E'	E	u = {~~B~~}	X

Visited sequence = { A, B, C, D, E }

DFS:

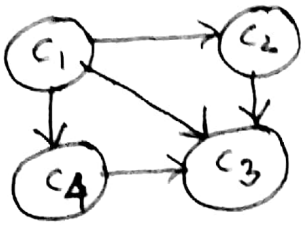


stack [A | | | | |]

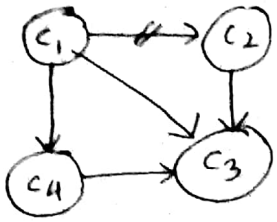
v ← stack	Visit v	Adjacent of v is u	stack ← v
v = 'A'	A	u = {B, C}	stack [B | C | | | |]
v = 'C'	C	u = {A, D}	stack [B | C | | | |]
v = 'D'	D	u = {B, C, E}	stack [B | C | E | | |]
v = 'E'	E	u = {D}	×
v = 'B'	B	u = {A, B}	×
v = 'B'	×	×	×

Visited Sequence = { A, C, D, E, B }

Topological Ordering



!



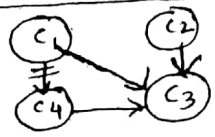
set 

$m \Leftarrow$ set $m = 'c_1'$

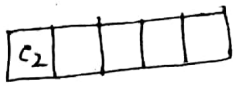
Order $\Leftarrow$ om Order $= \{c_1\}$

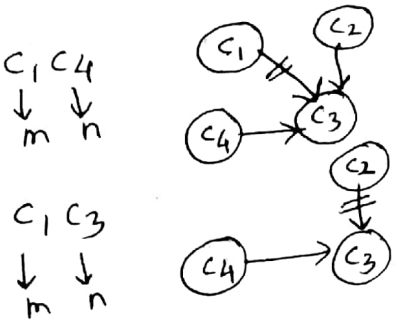
mn

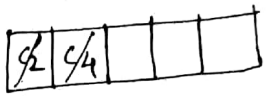
Discard mn



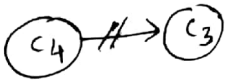
set $\Leftarrow$ n

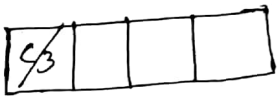
set 



set 

$m = 'c_2'$ order $= \{c_1 c_2\}$ $c_2 c_3$



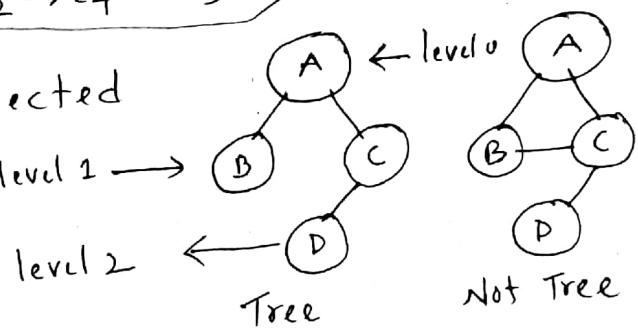
set 

$m = 'c_4'$ order $= \{c_1 c_2 c_4\}$ $c_4 c_3$ empty

$m = 'c_3'$ order $= \{c_1 c_2 c_4 c_3\}$ × ×

Topological order = $c_1 \rightarrow c_2 \rightarrow c_4 \rightarrow c_3$

Tree: Acyclic graph and Connected



Preorder: (root - left - right)

A B C D

~~Inorder: (root left right)~~

Inorder: (left - root - right)

B A D C

Postorder: (left - right - root)

B D C A

Level order

A B C D

height = max(level i)

= 2

children A is B, C
" C is D
A root of the tree

A is ancestor of D
leaf node B, D

father of B, C is A
" of D is C
B, C siblings