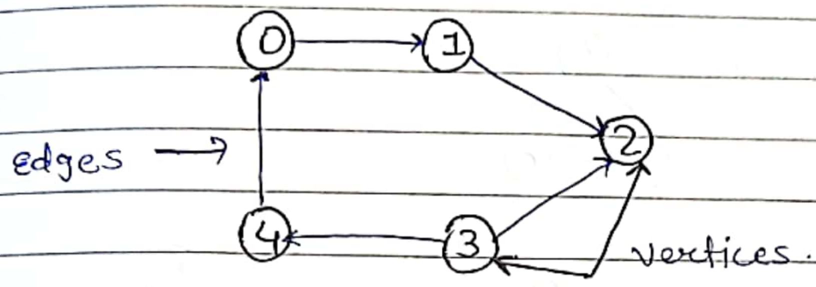
* DAA - N.P.P.

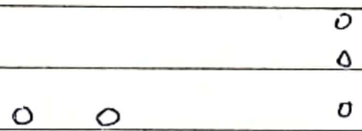
* Graph.

→ A graph is non-linear data structure consisting of nodes and Edges.



* Types of graph :-

1) Null graph :-

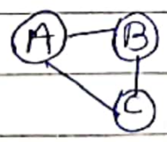


n=2 n=3

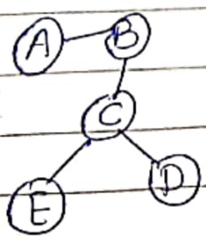
→ A graph which have no Edges betⁿ vertices Is called Null graph.

2). Simple graph :-

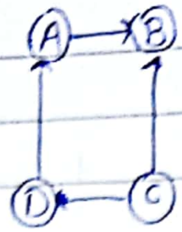
→ A simple graph is the undirected graph.



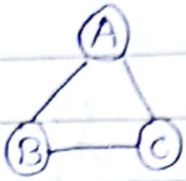
3). Undirected :-



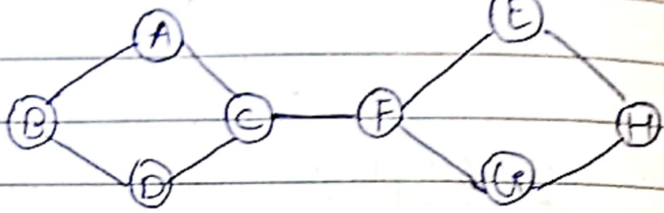
4). directed graph:-



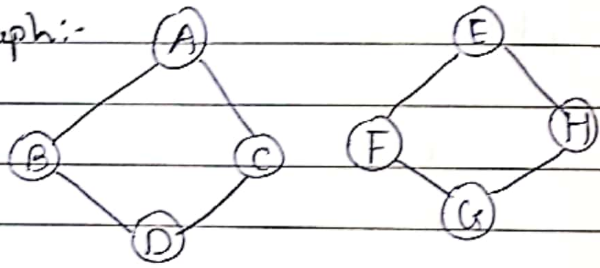
5). Complete graph:-



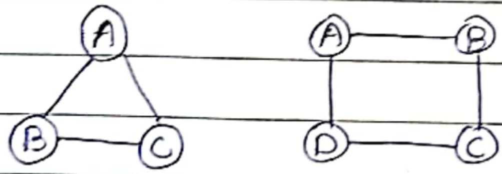
6). Connected graph:-



7). disconnected graph:-

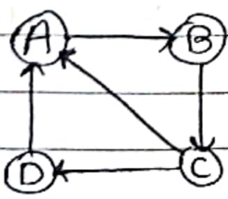


8). Regular graph:-



* Graph Representation:-

(1). Adjency matrix:- An, Example consider the directed graph.



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

(2). Adjancy list :-

A → B → D

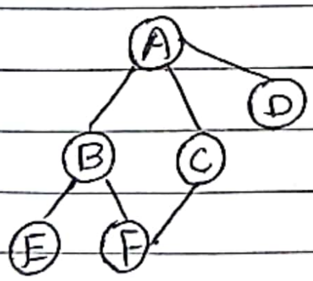
B → C

C → A → D

D

* Graph Traversal :

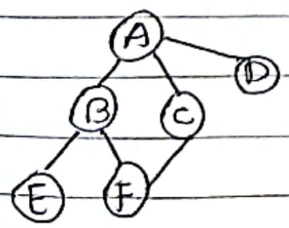
* BFS :-



→ BFS :- ABCDEF.

→ T.C :- O(V+E)

* DFS :-



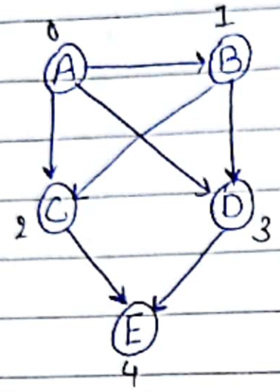
→ DFS :- ABEFCD.

→ T.C :- O(V+E).

* Topological sort :-

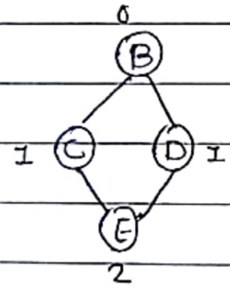
→ Find the Number of different topological orderings possible for given graph.

Step:-1



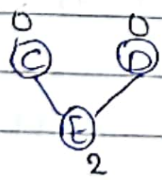
→ Write degree of Each vertex.

Step:-2 Vertex-A has least in degree
remove vertex-A and it's associated vertex.



A → B

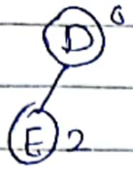
Step:-3 vertex-B has least in degree.
remove vertex-B and it's associated vertex.



→ Now, C and D both have least degree so, we remove vertex in DFs order.

→ DFS :- ABCDE

ep:3 → In DFS order 'C' comes ahead of 'D' so we remove C.



A → B → C

ep:4 → Now, we remove D

E

A → B → C → D

ep:5 → Now, we remove E

A → B → C → D → E

~~Topological order.~~

+ Examples:-

(1). $p = \{3, 4, 5, 6\}$
$w = \{2, 3, 4, 5\}$

→

i	w	p
1	2	3
2	3	4
3	4	5
4	5	6

p	w	i	0	1	2	3	4	5
		0	0	0	0	0	0	0
3	2	1	0	0	3	3	3	3
4	3	2	0	0	3	4	4	7
5	4	3	0	0	3	4	5	7
6	5	4	0	0	3	4	5	7

→ object 1 and 2.

* $V[i,W] = \max\{ V[i-1,W], V[i-1, W-w[i]] + P[i] \}$

→ $V[2,3]$

$V[1,3]$

$\max\{3,4\}$

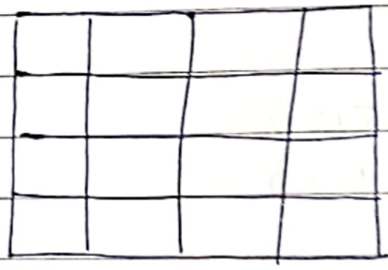
→ Ans:- {4}

* Back Tracking :-

* For Queens problem :-

→ rows
→ columns
→ diagonals

are not intersect Each-other



→ Total solution :- ${}^{16}C_4$

Ex:-

		q_1	
q_2			
			q_3
	q_4		

	q_1		
			q_2
q_3			
		q_4	

mirror image.

* Greedy Algorithms.

Ex:- profit = Pi {1, 2, 5, 6}
weight = ω {2, 3, 4, 5}

			0	1	2	3	4	5	6	7	8
p	ω	0	0	0	0	0	0	0	0	0	0
1	2	1	0	0	1	2	1	1	1	1	1
2	3	2	0	0	1	2	2	3	3	3	3
5	4	3	0	0	1	2	5	5	6	7	7
6	5	4	0	0	1	2	5	6	7	7	8

(1). fractional knapsack problem / knapsack problem

→ one wants to pack n items in a Bag:-

→ The ith item is worth vi rupees and weight wi kg maximize the value but cannot Exceed n kg vi, wi, ω are integers.

→ 0-1 knapsack → Each item is taken or not taken

→ fractional knapsack → fractions of items can be taken

que:-1

Object	1	2	3	4	5
Profit	5	2	2	4	5
weight	5	4	6	2	1
P/ω	1	0.5	0.33	2	5

capacity = 12

→ steps:- (1). arrange all object.
(2). arrange all profit.
(3). arrange all weights.
(4). find respective P/ω.

soln:

object	weight	profit
5	12-1 = 11	5
4	11-2 = 9	4
1	9-5 = 4	5
2	4-4 = 0	2
		16.

que:-2)

object	1	2	3	4	5
profit	30	20	100	90	160
weight	5	10	20	30	40
P/w	6	2	5	3	4

Capacity = 60.

soln:

object	weight	profit.
1	60-5 = 55	30
3	55-20 = 35	100
5	35-35 = 0	35×4 = 140
		270

que:-3)

object	1	2	3	4	5	6	7
profit	10	5	15	7	6	18	3
weight	2	3	5	7	1	4	1
P/w	5	1.66	3	1	6	4.5	3

soln:

object	weight	profit.
5	15-1 = 14	6
1	14-2 = 12	10
6	12-4 = 8	18
7	8-1 = 7	3
3	7-5 = 2	15
2	2-2 = 0	2×1.66 = 3.33
		55.33