Name -> Adhyan Dhyani
Class -> 4th sem, Sec-H
Roll no -> 20

Date.
Page No.

DAA- Tutorial - 5

Q) Ans)

	BFS	DFS
i)	Uses queue data structure.	Uses stack data structure.
ii)	Stands for Breadth first search	Stands for depth first search.
iii)	Can be used to find single source shortest path in an unweighted graph.	We might traverse through more edges to reach a destination vertex.
iv)	Siblings are visited before the children.	Children are visited before the siblings.
	Applications :	
1)	Shortest path & Min spanning tree for unweighted graph	Dectecting cycle in a graph.
2)	Peer to Peer network	Path finding
3)	Social Networking web.	Topological sorting
4)	GPS Navigation systems.	Solving puzzels with only one sol^{n}.

Ans 2
Soln In BFS we use Queue data Structures as queue is used when things don't have to be processed immediately, but have to be processed in FIFO order like BFS.

In DFS stack is used as DFS uses backtracking. For DFS we retrieve it from root to the farthest node as much as possible same as LIFO.

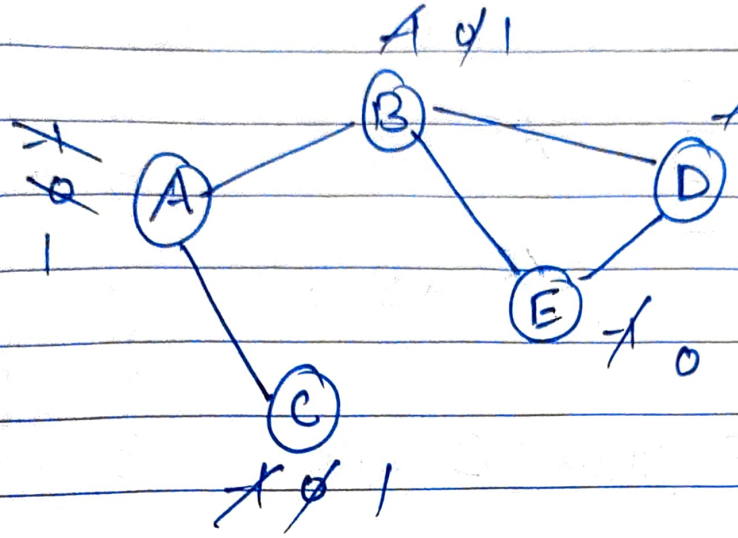
Ans 3
Soln Dence graph is a graph in which the no. of edges is close to the maximal no. of edges

Sparse graph is a graph in which the no. of edges is close to the minimal no. of edges. It can be disconnected graph

Ans 4
Soln

Date. _____
Page No. _____

Cycle detection in undirected Graph (BFS)



- => unvisited
0 => into the queue.
1 => traversed

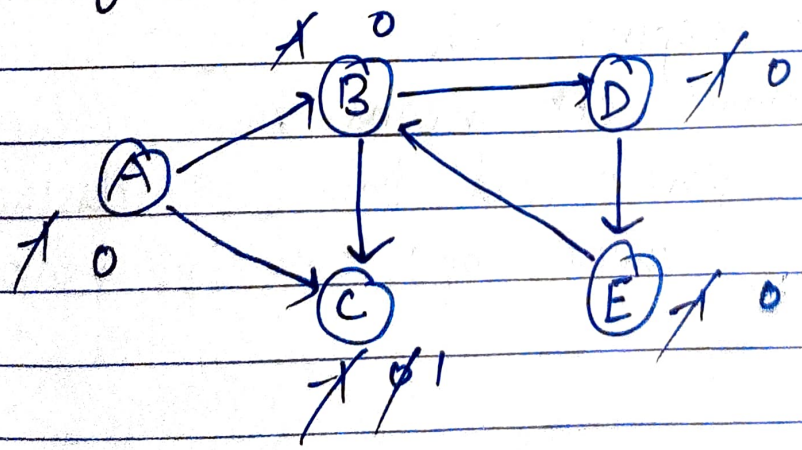
Queue :

A	B	C	D	E

Visited Set :

A	B	C	D

Cycle detection in Directed Graph (DFS)



- => unvisited
0 => visited & in stack.
1 => visited & popped out.

Date.
Page No.

Stack :

Visited set

ABCDE

B → D → E → B

Parent map

Vertex	Parent
A	—
B	A
C	B
D	B
E	D

Here E finds B with 0,

→ it contains a cycle.

Ans 5

Sol^n The dis-joint set data structure is also known as union-find data structure & merge-find set. It is a data structure that contains a coll^n of disjoint or non-overlapping sets.

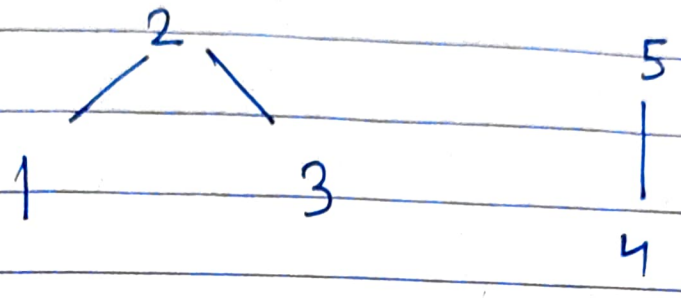
The disjoint set means that when the set is partitioned into the disjoint subsets, various op^n can be performed on it.

Date.
Page No.

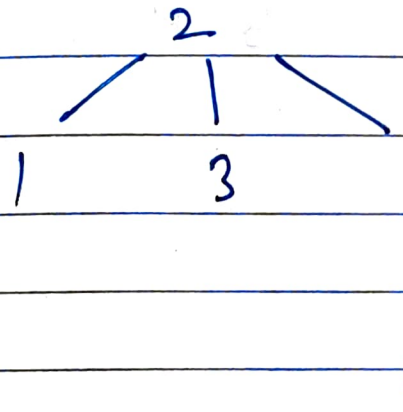
Operations on disjoint set.

1) Union.

Eg. S1 S2



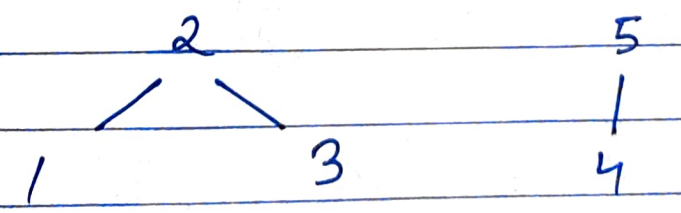
S1 ∪ S2



Merge the sets containing x & containing y into one.

2) Find

Eg. S1 S2



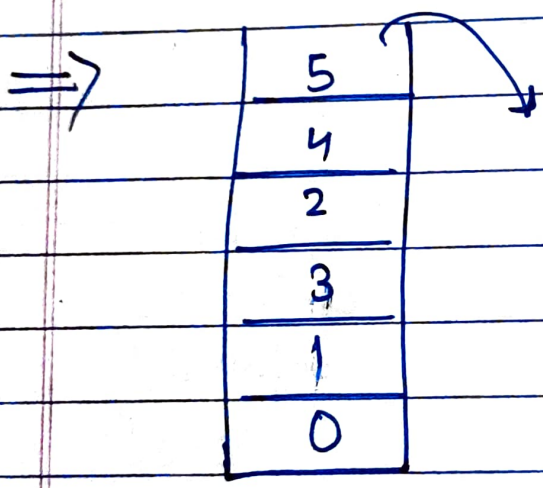
find (3) => S1
find (5) => S2

return in which set x belongs.

Date.
Page No.

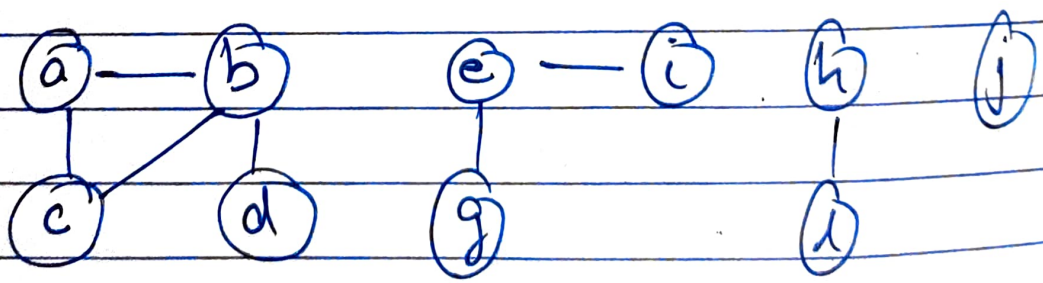
3) Make-Set (X) : Create a set containing X.

Ans6.
Sol^n Go to node 5, all its adjacent nodes are already visited so push node 5 into the stack & mark it visited.



5 4 2 3 1 0

Ans7
Sol^n



$V = \{a, b, c, d, e, g, h, i, j, l\}$

$E = \{(a,b), (a,c), (b,e), (b,d), (e,i), (e,g), (h,l), (j)\}$

Date.
Page No.

	{a}{b}{c}{d}{e}{g}{h}{i}{j}{l}
(a,b)	{a,b}{c}{d}{e}{g}{h}{i}{j}{l}
(a,c)	{a,b,c} d, e, g, h, i, j l
(b,c)	{a,b,c} d, e, g, h, i, j, l
(b,d)	{a,b,c,d} e, g, h, i, j, l
(e,i)	{a,b,c,d}{e,i} g, h, j, l
(e,g)	{a,b,c,d}{e,i,g} h, j, l
(h,l)	{a,b,c,d}{e,i,g}{h,l}{j}
(j)	{a,b,c,d}{e,i,g}{h,l}{j}

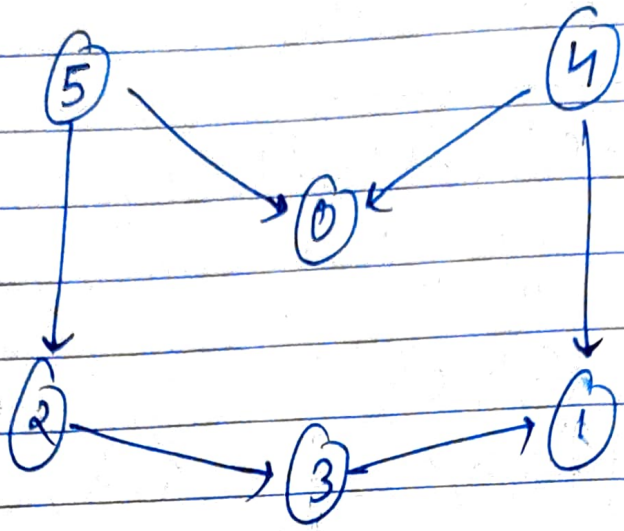
We have,

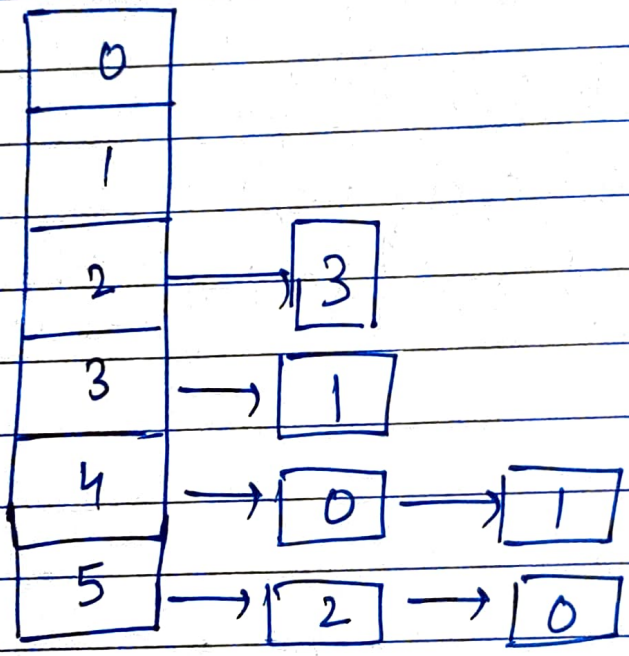
{a, b, c, d}
{e, i, g}
{h, l}
{j}

Date.
Page No.

Ans 8

Sol^n





Algo :-

1) Go to node 0, it has no outgoing edges so push node 0 into the stack & mark it visited.

Date.
Page No.

2) Go to node 1, again it has no outgoing edges, so push node 1 into the stack & mark it visited.

3) Go to node 2, process all the adjacent nodes & mark node 2 visited.

4) Node 3 is already visited so continue with next node.

5) Go to node 4, all. ith adjacent node are already visited so push node 4 into the stack & mark it visited.

Ans 10
Soln

Min Heap	Max Heap.
1) For every pair of the parent decendant child node, the parent node always has lower value.	For every pair of the parent & decendant child node has greater value.

Date.
Page No.

2)	The value of nodes including as we traverse from root to leaf node.	The value of nodes decreases as we traverse from root to leaf node.
3)	Root node has the lowest value.	The root node has the greatest value.