Name: T Rishita Reddy
Sec: I
Roll no: 47

Date : Page :
Topic :

Tutorial-5

Q:1

Ans⇒

	BFS	DFS
1.	Uses queue data structure.	Uses stack data structure.
2.	Stands for Breadth First Search.	Stands for Depth First Search.
3.	Can be used to find single source shortest path in an unweighted graph, & we reach a vertex with min. no of edges from a source vertex.	We might traverse through more edges to reach a destination vertex from a source.
4.	Siblings are visited before the children.	Children are visited before the siblings.
	Applications:	Applications:
(i)	Shortest Path & Minimum Spanning Tree for unweighted graph.	(i) Detecting cycle in a graph.

Signature .....

STUDENT STYLE

Date : Page :
Topic :

(ii) Peer to Peer Networks (ii) Path finding.

(iii) Social Networking Websites (iii) Topological Sorting.

(iv) GPS Navigation system. (v) Solving puzzles with only one soln.

Q.2

Ans⇒ In BFS we use Queue data structure 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, this is the same idea as LIFO [used by stack].

Q.3

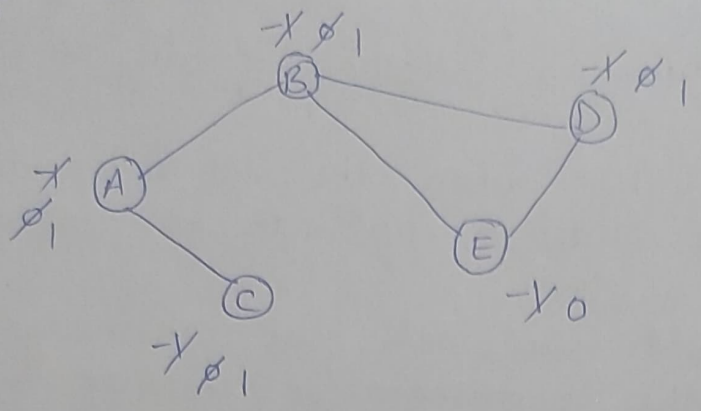
Ans⇒ Dense graph is a graph in which the no. of edges is close to the maximal no. of edges.
Sparse graph is a graph in which the no. of edges is close to the minimal no. of edges. It can be disconnected graph.
* Adjacency lists are preferred for sphere graph & Adjacency matrix for dense graph.

Signature.....

STUDENT STYLE

Q.4

Ans ⇒ Cycle detection in undirected Graph (BFS)



-1 = Unvisited

0 = into the queue (node)

1 = traversed.

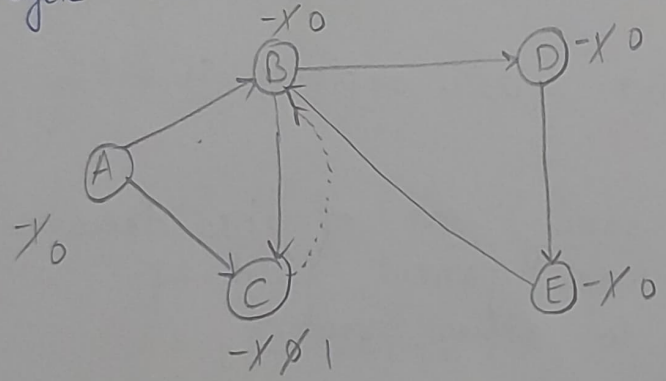
Queue: | A | B | C | D | E |

Visited Set: | A | B | C | D |

when D checks it adjacent vertices it finds E with 0.

→ If any vertex finds the adjacent vertex with flag 0, then it contains cycle.

Cycle Detection in Directed Graph (DFS)

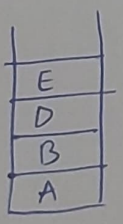


-1 = unvisited

0 = visited & in stack

1 = visited & popped out from stack.

Stack:



Visited Set:

ABCDE

⇒ B → D → E → B

Here E finds B (adjacent vertex of E) with 0.

→ it contains a cycle

Parent Map

Vertex	Parent
A	–
B	A
C	B
D	B
E	D

Q.5

Ans→ The disjoint set data structure is also known as union-find data structure & merge-find set. It is a data structure that contains a collection of disjoint or non-overlapping sets.

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

In this case, we can add new sets, we can merge the sets, & we can find the representative member of a set. It also allows to find out whether the two elements are in the same set or not efficiently.

Operations on disjoint set.

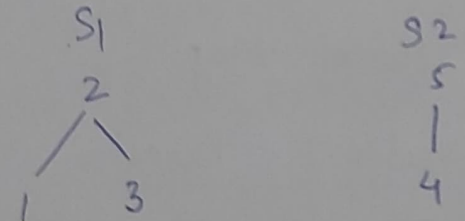
① Union

(a) If S1 & S2 are two disjoint sets, their union S1 ∪ S2 is a set of all elements x such that x is in either S1 or S2.

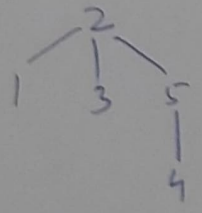
(b) As the sets should be disjoint S1∪S2 replaces S1 & S2 which no longer exists.

(c) Union is achieved by simply making one of the trees as a subtree of other i.e. to set parent field of one of the roots of the tree to other root.

Ex:



S1 ∪ S2

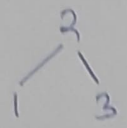


Merge the sets containing x & containing y into one.

2. Find

Give an element x, to find the set containing it

Ex: S1



S2

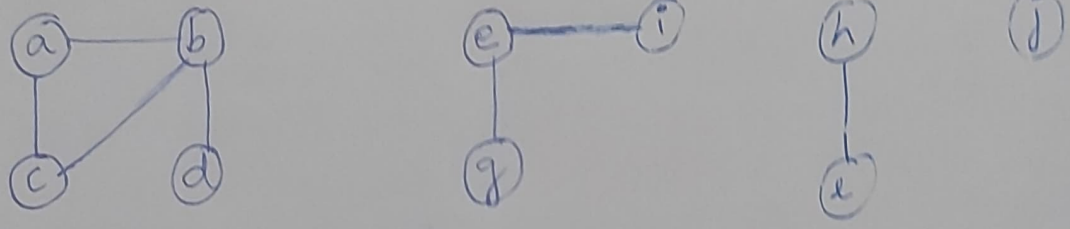
find(3) → S1
find(5) → S2

return in which set x belongs

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

Q.7

Ans →



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

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

	{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}	