Name: Shambhavi Vats.

Date.______

Page No.______

Tutorial-5

Q1 What is the difference b/w DFS and BFS. Applications of Both Algo.

BFS	DFS
→ Stands for Breadth first Search.	DFS stands for Depth first Search.
→ BFS is more suitable for searching vertices which are closer to source.	DFS is more suitable for searching vertices away from source.
→ BFS considers all neighbours first, so not suitable for descision making trees in games/puzzles.	In DFS we make a descision, then explore all the paths through this descision, if descision = win, stop, therefore suitable in games/puzzles.
→ Siblings are visited before children.	children are visited before sibling.
→ Implemented using FIFO list.	Implemented using LIFO list.
→ Requires more m/m as compered to DFS.	Requires less m/m.
→ No Backtracking required.	Need of Backtracking
→ Slower than DFS.	Faster than BFS
→ O(V+E) V=vertices, E=Edges	O(V+E)
→ visited according to tree level.	visited according to tree depths.

Date.
Page No.

Application of BFS -
→ Peer to Peer Networks
→ Web Crawlers
→ Network Broadcasting.

Application of DFS -
— Detecting cycle in graph
→ Topological sort.
— Solving puzzles with only one solution

Q2 Which Datastructure is used to implement and Why?
→ BFS → queue
DFS → stack

Q3. What do you mean by sparse and dense graphs? Which representation of graph is better for sparse and dense graphs?

Dense graph is a graph in which the no of edges is close to the maximal no of edges. Adjacency matrix is preferred.

Sparse graph is a graph in which no of edges is close to the minimal no of edges.
sparse graph can be a disconnected graph
Adjacency list is required.

Date.
Page No.

Q4 How can you detect a cycle in a graph using BFS and DFS?
DFS :
DFS for a connected graph produces a tree.
There is a cycle in a graph only if there is a back edge present in graph.
↳ An edge that is joining a node to itself (self loop) or one of its ancesto in the tree produced by DFS.

BFS : For every visited vertex 'v' if there is an adjacent 'u' such that u is already visited and u is not a parent of v, then there is a cycle in graph. If no adjacent of any vertex is found, then there is no cycle.

Q5 What do you mean by disjoint ds? Explain three operations along with examples which can be performed on disjoint sets.
→ Also known as union find data structure or merge find data structure. that keeps track of set of elements partitioned into a number of disjoint subsets.

Disjoint set can be defined as the subsets where

Date.________
Page No.________

there is no common element b/w the two sets.

operations -!

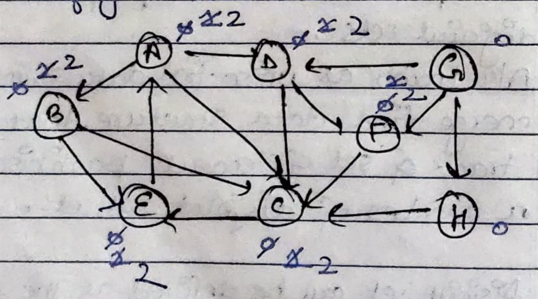
① Making new sets
The MakeSet operation adds a new element
The element is placed into a new set
The element and new set is containing only new element and new set is added to ds.

② Finding the representative of set containing a ~~new~~ given element
can be implemented by recursively traversing parent array until we hit a node who is parent to itself.
"Find".

③ Union - It takes two elements a input and finds the representatives using "find" operation and puts either one of the trees under the root node of other tree, merging the trees and sets.

Q6

BFS

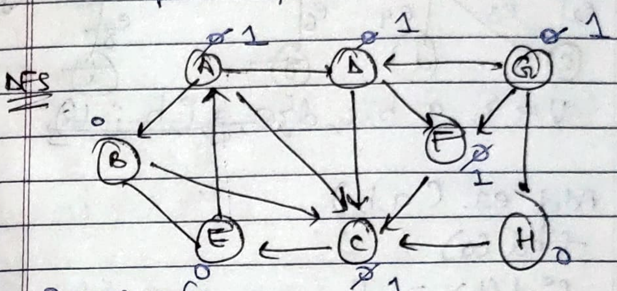


Date.________
Page No.________

Node	~~A~~	~~B~~	~~C~~	~~D~~	E	F		
Parent	–	A	A	A	B	D		

* No path b/w A and H.

DFS



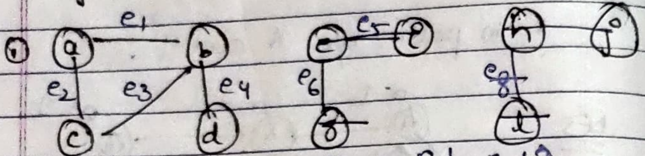
Source - G
Dest - A

Node proc	Stack
	G
G	D F H
~~G~~ D	~~E C F H~~ ~~F H F C~~
F	C C F H
C	E C F H
E	A C F H
A	B C F H

$G \rightarrow D \rightarrow F \rightarrow C \rightarrow E \rightarrow A$

Date.
Page No.

Q7 Find out no. of connected components and vertices in each component using disjoint set Data Structure.

①


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

① Add e1 (a, b)
find (a) = U
find (b) = U
$S_1 = \{a, b\}$
U = { c, d }

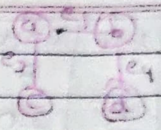
③ Add e_2 (a, c)
find (a) = S_1
find (c) = U
$S_1 = \{a, b, c\}$
U = {d}

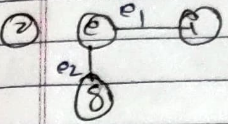
⑤ Add e_3 (b, c)
find (b) = S_1 } cycle detected
find (c) = S_1 }

④ Add e_4 (b, d)
find (b) = S_1
find (d) = U

Date.
Page No.

$S_1 = \{a, b, c, d\}$
U = {φ}

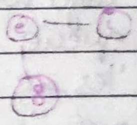


②


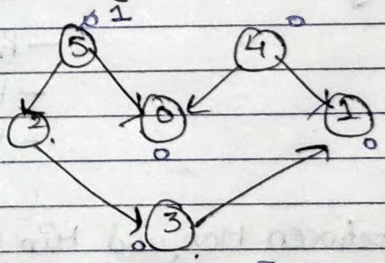
U = {e, i, g}

① Add e_1 (e, i)
find (e) = U , find (i) = U
$S_1 = \{e, i\}$
U = {g}

② Add e_2 (e, g)
find(e) = S_1
find (g) = U
$S_1 = \{e, i, g\}$



Q8 Apply topological sorting and DFS on graph having :



① Topological sort = 5, 4, 0, 2, 3, 1 .
② DFS

Date.
Page No.

Node	Stack
	5
5	20
2	30
3	10
1	0

5→2→3→1

Q9. Heap ~~DS~~ can be used to implement priority queue? Name few graph algo where you need to use priority queue and why?

Yes, we can use heaps to implement the priority queue.
O(nlogn) to insert and delete.

Priority queue is used in Dijkstra's Algo
— Prim's Algo
— Huffman code
— heap sort

Q10 Diff between Max and Min Heap?

Min Heap	Max Heap
In Min Heap, key ~~is~~ present at the root node. and it must be <= all keys present of his children	key present ~~is~~ present at root must be >= keys present all all of its children.
→ Min key element is present at the root.	Maximum key element present at root.
→ Ascending priority.	Descending priority
→ Smallest element is popped first.	largest element is popped first.
→ Get Maximum Element, Extract Maximum and insert.	Get Minimum, Extract Minimum and Insert.

Date.
Page No.