Tutorial-5

Q1

Key	BFS	DFS
definition	Stands for Breadth first search	Stands for depth first search
Data Structure	It uses Queue to find shortest path	It uses stack to find shortest path
Source	It is better when target is closer to source	It is better when target is far from source
Suitable for decision tree	It consists all neighbour so it is not suitable for decision tree	It is more suitable with decision.
Speed	It is slower than BFS	It is faster than BFS

Q2 Stack is used to implement DFS, because in it we first traverse the whole branch of tree and later on visit the adjacent branch since this is similar to LIFO therefore stack is used.

Queue is used to implement BFS it is because queue is used as FIFO instead because BFS is to test the immediate children first and after all immediate children are tested to then return to these children & check their children & so forth.

Q3 Sparse graph — Graph where no. of edges is much less than possiblee no. of edges

Dense graph — where no of edges is much close to maximal no. of edges

If graph is dense it should be represented by adjaacency matrix

If graph is sparse it should be represented by adjacency list

Q4 BFS

In undirected graph do a BFS traversal on given graph for each visited vertex v, if there is an adjacent 'u' such that 'u' is already visited & 'u' is not parent of 'v' then there is cycle in graph

DFS

run DFS from a node & mark this node as visited now for any other vertex if its neighbour is already visited & that neighbour is not parent of that current node then there exist a cycle in graph

Q5 Disjoint set data structure

The disjoint set ~~d~~ can be defined as subsets where there is no common element b/w 2 sets.

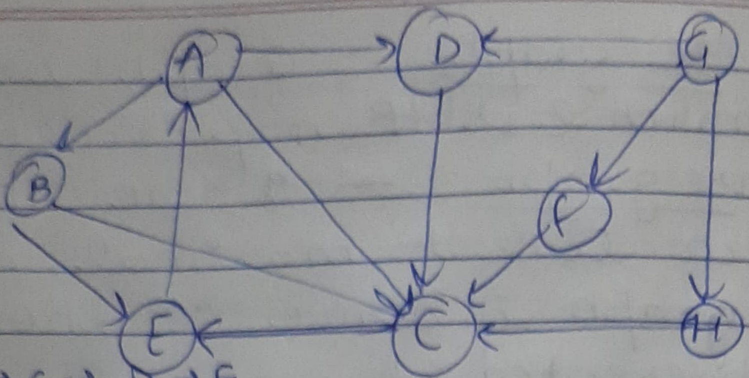
Operations are

i) Union

ii) Make new set

iii) find

Q6 BFS.



A → B → C → D → E
G → H → F

DFS
A → D → C → B , G → F → H

Q7 Connected components = 4
Vertices = 10

Q8 Topological sort → 0 – 1 – 2 – 3 – 4 – 5
DFS → 5 → 2 → 3 → 1 → 0
4 can't be reached

Q9 Yes, heap data structure can be used to create priority queue
- Dijkstra's to find shortest path
- Prim's Algo
- Hoffman Algo

Q10 Min Heap → root element is smallest
Max Heap → root element is largest