Tutorial 5 DAA

Q-1)

Ans:

	BFS	DFS
Definition	Stands for Breadth First Search	Stands for Depth First Search
DataStructure	It uses Queue for finding the shortest path	It uses stack to find the shortest path
Sources	It is better when target is closer to source.	It is better when target is far from source.
Decision Tree	It consider all neighbour so it is not suitable for decision tree used in puzzle game.	It is more suitable as with one decision. we need to traverse further to augment the decision
Speed	It is slower than DFS	It is faster than BFS
Time Complexity	O(V+E) where V is vertices and E is edges.	O(V+E).

Q-2)

Ans: Stack is used to implement DFS, because in it we first traverse the whole branch of the 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 a FIFO instead because BFS is to test the immediate children first and after all immediate children are tested, to then return to those children and check their children and so forth.

Q-3)

Ans: Sparse Graph: Graph where no. of edges is much less than the possible no. of edges.

Dense Graph: where no. of edges is much more close to maximam no. of edges.

If graph is dense, it should be represented by adjancey matrix.

If graph is sparse, it should be represented by adjacency list.

Q-4)

BFS

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

DFS

Run DFS from a node and mark this node as visited, now for any other vertex if it neighbour is already visited and that neighbour is not the parent of that cussent node then there exist a cycle in the graph.

Q-5)

Ans: Disjoint Set Data Structure

The disjoint set can be defined as the subsets where there is no common element blw two sets.

Operations are:

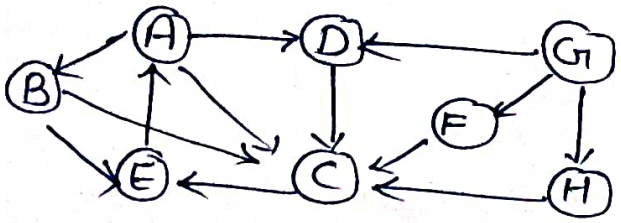
i) Union
ii) Make new set
iii) find

Q-6

Ans:

BFS

A → B → C → D → E,
G → H → F



DFS

A → D → C → B, G → F → H

Q-7

Ans. Connected Components → 4
Vertices = 10

Q-8

Ans.. Topological sort → 0-1-2-3-4-5

DFS → 5 → 2 → 3 → 1 → 0

4 can't be reached

Q-9)

Ans: Yes, heap data structure can be used to create priority queue.

- Dijkstra's to find shortest path.
- Prism's Algo.
- Hoffman Algo.

Q-10)

Ans: Min. heap → root element is the smallest.
Max. heap → root element is the largest