Name - Ayush Chauhan

Sec - B

Roll No. - 37

Date: ___/___/___

Tutorial - 5

1.

	BFS	DFS
(i)	It stands for Breadth first Search	It stands for Depth first Search.
(ii)	It uses Queue Data Structure.	It uses stack data structure.
(iii)	It is more suitable for searching vertices which are closes to given source	It is more suitable when there are solutions away from source.
(iv)	Here, siblings are visited before children	Here children are visited before siblings
(v)	There is no concept of backtracking.	It is a recursive algo. that uses backtracking
(vi)	It requires more memory.	It requires less memory.

Applications:

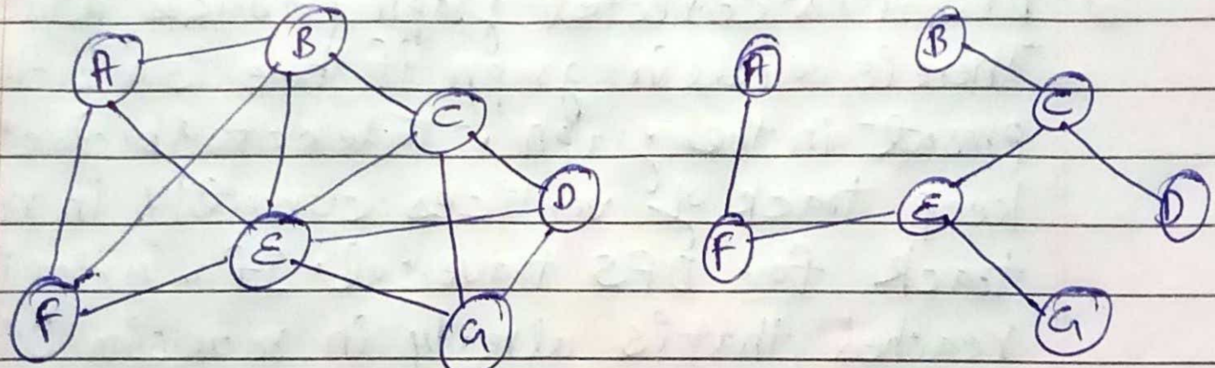
(1) BFS - Bipartite graph and shortest path, peer to peer networking, GPS navigation system.

(2) DFS - Acyclic graph, topological sort, scheduling problems.

2. For implementing BFS, we need a queue data structure for finding shortest path b/w any node. We use queue because things don't have to be processed immediately, but have to be processed in FIFO order.

for implementing DFS, we need a stack data structure as it traverse a graph in depthward motion and uses stack to remember to get the next vertex to start a search, when a dead end occurs in any iteration

3. Sparse graph is a graph in which no. edge is very less.
Dense graph is a graph in which in which no of edge is close to maximal no of edges.



Dense Graph Sparse Graph

For sparse graph, it is preferred to use Adjacency list.
for dense graph, it is preferred to use Adjacency matrix.

4. For detecting cycle in BFS:
Use Kahn's algorithm for Topological Sorting —
1. Compute in-degree for each of vertex present in graph & initialize count of visited nodes as 0.
2. Pick all vertices with in-degree as 0 and add them in queue

3. Remove a vertex from queue and then
   - increment count by 1.
   - If in-degree of neighbouring node is reduced to 0 then add to queue.
4. Repeat (3) until queue is empty.
5. If count of visited nodes is not equal to no. of nodes in graph has cycle, otherwise not.

For detecting cycles in DFS:

DFS for a connected graph produces a tree. There is cycle in graph if there is a back edge present in the graph. To detect a back edge, keep track of vertices currently in recursion track for DFS traversal. If a vertex is reached that is already in recursion stack, then there is a cycle.

6. BFS:

Child	G	H	D	F	C	E	A	B
Parent		G	G	G	H	C	E	A

DFS:

Nodes Visited: ~~G~~, D, H, ~~F~~, ~~C~~, ~~E~~, ~~A~~, ~~B~~

Stack: G, F, C, E, A, B

Date : ___/___/___

7. V = {a} {b} {c} {d} {e} {f} {g} {h} {i} {j}

E = {a,b} {a,c} {b,c} {b,d} {e,f} {e,g} {h,i}
{j}

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

No. of connected components = 3

8. We take source node as 5
Applying Topological Sort

DFS(5) q: 5/4, POP 5
↓
DFS(0) q: 4/2, POP 4
↓
DFS(2) q: 2/0, POP 2
↓
DFS(3) q: 0/3, POP 0, POP 3
↓ q: 1, POP 1
DFS(1) Answer: 5 4 2 0 3 1

10.	Min Heap	Max Heap
1.	Key present in root node must be less than or equal to among keys present at all of its children.	Key present at root node must be greater than or equal to among keys present at all of its children
2.	It uses ascending priority.	It uses descending priority.
3.	The smallest element is first to be popped from the heap.	The larger element is the first to be popped from the heap.