Gagan T
24UG00811

BFS & DFS :

Algorithm

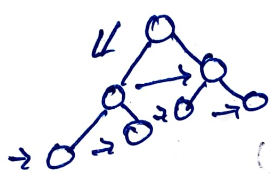
1) BFS : Breadth First Search

→ Explores level by level.
→ First visits Neighbours then goes to child Nodes.
→ Finds the shortest path if possible.

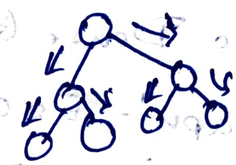
DFS: Depth first search.

→ Explores depth first then neighbours
→ Uses regression to track explored paths.
→ Does not guarantee shortest path.

(2) BFS



DFS



DFS: [LIFO] STACKS

Data Structure: Queue[FIFO]

3) ~~DFS &~~ Time Complexity

* Both DFS & BFS must visit all vertices (N) & traverse all edges (E)
+ Each vertex is enqueued or pushed once and dequeued or popped once.

+ Each edge is checked at most twice
[twice = undirected graphs].

Time Complexity :- O[N+E]

4) Space Complexity

a) BFS

- Needs a queue that can hold upto (N) nodes in the worst case

* Also requires adjacency list storage :- $O(N+E)$

* Space :- $O(N+E)$

b) DFS :

→ Needs recursion stack

→ In worst case, regression depth = $O(N)$

→ plus adjaceney list : $O(N+E)$

→ Space :- $O(N+E)$

5) Sparse v/s Dense Graphs.

a) Sparse Graph :- $E \approx O(N)$

- Complexity :- $O(N+E) \simeq O(N)$

b) Dense Graph : $E \simeq O(N^2)$

- Complexity : $O(N+E) \simeq (N^2)$