Derivates of BFS & DFS

1) Algorithm Explores the graph

→ (BFS) :- BFS Explores the graph level by level. it begins at a source vertex, visit all neighbours. then moves to neighbour to those neighbours & continue until all reachable vertices as visited.

→ (DFS) :- Explores the graph in depth-oriented manner, starting from a source. it follows one path as possible before backtracking and Explores univisted vertices.

2) Data Structure used :-

→ BFS :- use a queue (FIFO) to maintain the order of vertices to be visited array/set to mark Explored vertices.

→ DFS :- use stack and a visited array/set to mark explored vertices.

3) Derivation of Time complexity

(BFS) :- Each vertex is enqueued and dequeued once → O(N) operation every vertex. all its adjacency list edges transvered Across the graph edge one → O(E) operations. hence time complexity O(N+E)

(DFS) :- Each vertex pushed popped from stack, most one → O(N) operation, each adjacency scanned once. & every edge Exactly once → O(E) operation hene space time complexity O(N+E)

4) Derivation of space complexity

(BFS) :- Queue store up to O(N) vertius, visited array requires O(N) and adjacency list requires O(N+E) total Space = O(N+E)

(DES) :- Recursion stack can go deep as O(N), visited array O(N) adjacency List requires O(N+E) total space = O(N+E)

5) Sparse VS Dense graph cowc

→ sparse graph (E = O(N)) → BFS and DFS take O(N)

→ Dense graph (E = O(N^2)) → BFS and DFS take O(N^2)