BFS and DFS Complexities

(1) How does BFS and DFS explore?

BFS: *BFS explore level by level i.e visiting all the neighbors of a node before going deeper.

*Uses queue [FIFO]

DFS: *Explores by going as deep as possible along a path, then backtracks.

*Uses stack or recursion.

2) Assumptions Made

* Graph is stored as adjacency list [memory-efficient]
* Graph is connected, hence every node can be reached from the starting.
* No parallel edges or self-loops for simplicity.
* All nodes and edges are included in the complexity.

(3) Data Structures Used

BFS: Uses a queue [FIFO] to keep track of the next node to visit.

DFS: Uses a stack [LIFO] or recursion.

(4) Time and space complexities

Let, N: Number of nodes
E: Number of edges

BFS: Time Complexity: O(N+E)

Because every node and every edge is visited at most once.

Space Complexity: O(N)

The largest queue size is proportional to the no. of nodes.

DFS

Time Complexity : O(N+E)

Visiting all nodes and all edges once

Space Complexity: O(N)

Maximum stack/depth or recursion depth is N.

(5) Sparse VS Dense Graphs.

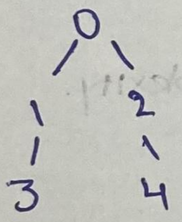
Sparse graph:

Fewer edges, so E ~~dose~~ close to N. Time and space are mostly spent on nodes.

Dense graph:

Many edges, E can approach N^2. Time and space also grow with more edges, but both BFS and DFS still run in O(N+E).

Example



BFS starting from 0

Order: 0 -> 1 -> 2 -> 3-4 (level by level)

DFS starting from 0

Order: 0 -> 1 -> 3 -> 2 -> 4 [goes deeper)