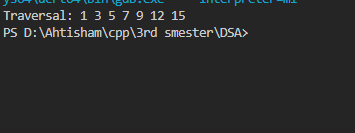
**1. Depth-First Search (DFS) in a Tree**

Depth-First Search is a method for traversing a tree structure where:

* The traversal begins at the root node.
* It explores each branch as far as it can go before returning to explore other branches.
* The approach is typically implemented using recursion.

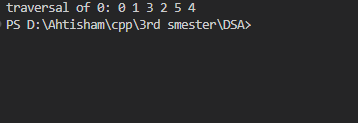
DFS is valuable for exploring all nodes in a tree and is commonly used in search and pathfinding algorithms



**2. Depth-First Search (DFS) in Graphs**

DFS in graphs functions similarly to its use in trees, but with some key differences:

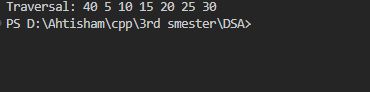
* A *visited* list is required to prevent revisiting nodes and getting stuck in cycles.
* Starting from a node, it recursively explores all adjacent, unvisited nodes.
* It can be applied to both directed and undirected graphs.



**3 BFS in Trees**

Breadth-First Search (BFS) is a tree traversal technique that processes nodes level by level:

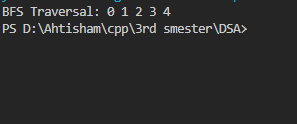
* It begins at the root and explores all nodes at one depth before moving to the next level.
* A queue (often a custom implementation) is used to manage the order of node visits.
* BFS is effective for finding the shortest path and for examining the tree layer by layer.



**4. Breadth-First Search (BFS) in Graphs**

**BFS in graphs also relies on a queue-based approach:**

* **It begins from a starting node and visits all its neighboring nodes before moving to the next level.**
* **A visited list is maintained to avoid revisiting nodes and forming cycles.**
* **Commonly used for level-order traversal and finding the shortest path in graphs without edge weights.**

****