|  |  |
| --- | --- |
| Project Title : | Lab Task 6 |
| Name : | **Aman Ali** |
| Roll No : | **195** |
| Class : | BSAI |
| Section : | **3C** |
| Submission Date : | September 25, 2025 |

Breath First Search

**Code Explanation**

**1. Recursive BFS (bfs\_recursive)**

* This function performs BFS using **recursion**.
* It accepts a list of nodes at the current level (level\_nodes) and a goal.
* For each node at the current level:
  + Prints the node and marks it as visited.
  + Checks if the node is the goal; if yes, stops the search.
  + Adds all children of the node to a next\_level list.
* The function **recursively calls itself** with the next\_level until all nodes are visited or the goal is found.

**2. BFS Using List as Queue (bfs\_list\_queue)**

* Implements BFS iteratively using a **list as a queue**.
* The queue stores nodes to be visited next.
* While the queue is not empty:
  + Pop the first node from the queue (FIFO order).
  + Print the node and mark it as visited.
  + If the node is the goal, stop the search.
  + Append all unvisited children to the queue for further exploration.

**How It Works**

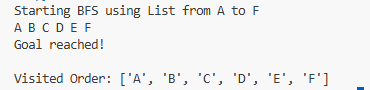
* BFS explores the tree **level by level**, starting from the root node.
* Both versions maintain a **visited list** to avoid processing nodes multiple times.
* Recursive BFS works by keeping track of **current level nodes** and calling itself for the **next level**, while the iterative version keeps nodes in a **queue** and processes them in **FIFO order**.
* Both approaches guarantee that the **shortest path to the goal** in terms of edges is found first in unweighted trees.

**Why This Approach**

* BFS ensures **complete traversal** of all nodes level by level.
* Using recursion simplifies understanding the concept of levels but may risk **stack overflow** on very large trees.
* Using a queue is **memory-efficient** and easier to implement for large trees or graphs.
* Both methods are **deterministic** and always find the goal if it exists.

**Sample Output**

**Recursive BFS**



**Iterative BFS Using List as Queue**

