Name: Saira Irshad

Roll no: SU92-BSAIM-f24\_121

Submitted To: Sir Rasik Ali

Section: BSAI-3b

**Breadth First Search (BFS) in Artificial Intelligence**

Breadth First Search (BFS) is a fundamental graph traversal algorithm widely used in Artificial Intelligence for exploring nodes and edges systematically. It follows the principle of visiting nodes level by level, ensuring that all neighbors of a node are visited before moving to the next level.

# 1. BFS with Queue

In the standard BFS algorithm, a Queue data structure is used to maintain the order of nodes to be visited. The process is as follows:  
1. Start with the root/start node and put it in the queue.  
2. Remove a node from the front of the queue, visit it, and mark it as visited.  
3. Add all unvisited neighbors of that node into the queue.  
4. Repeat until the queue is empty.  
  
This ensures that BFS explores nodes level by level. Queue guarantees FIFO (First In, First Out) order.

✔ Advantages of BFS with Queue:

* • Simple and widely used approach.
* • Guarantees shortest path in unweighted graphs.
* • Easy to implement using built-in data structures.

❌ Disadvantages:

* • Requires extra memory for maintaining the queue.
* • Can be slow if the graph is very large.

# 2. BFS without Queue

Although BFS is naturally implemented using a Queue, it can also be achieved without an explicit queue. This is done by using recursion and lists to keep track of the current level and the next level. The idea is:  
1. Start with a list containing the root node.  
2. Process all nodes at the current level and collect their unvisited neighbors.  
3. Pass the neighbors list recursively as the next level.  
4. Continue until no nodes remain.  
  
This approach replaces the queue with recursion and level lists, but the working principle remains the same.

✔ Advantages of BFS without Queue:

* • No explicit queue data structure is needed.
* • Code can be more readable using recursion.

❌ Disadvantages:

* • Recursive calls may increase memory usage (stack).
* • Slightly less efficient than queue-based BFS.

# Conclusion

Both versions of BFS (with Queue and without Queue) serve the same purpose of traversing graphs level by level. The Queue-based method is the standard and more efficient implementation, while the recursive version without Queue provides an alternative approach that may be easier to understand in some cases.