BFS

(Breadth-First Search)

1.What BFS Means :

BFS = Breadth-First Search
It means we explore a graph **level by level**, starting from one node.

1. First visit all the **closest** nodes
2. Then visit the next-level nodes
3. And so on…

Just like checking people **row by row** in a classroom.

2.Why BFS Uses a Queue :

A **queue** works like a line:

1. First person goes first
2. Next person goes next

This is **FIFO (First In First Out)**.

BFS uses a queue to make sure:

1.Nodes added **first** are visited **first**

2.Which makes traversal **level by level**

3.Diagram:

```
  A

 / \

B   C

|   |

D   E
```

Connections:

A → B, C

B → D

C → E

Now let’s visualize BFS starting from **A**:

```
LEVEL 0:      A

             / \

LEVEL 1:    B    C

            |    |

LEVEL 2:    D    E
```

This is why BFS goes **A → B → C → D → E**

4. BFS Traversal Example:

Queue Step-by-Step

Start with A:

Queue: [A]
Visited: A
Output: A

Look at A’s neighbors → B, C

Queue: [B, C]
Visited: A, B, C
Output: A

Now take B:

Queue: [C, D]   (because B → D)
Output: A B

Now take C:

Queue: [D, E]   (because C → E)
Output: A B C

Take D:

Queue: [E]
Output: A B C D

Take E:

Queue: []
Output: A B C D E

5.BFS Pseudocode:

```
BFS(start):

    create a queue Q

    mark start as visited

    enqueue start into Q



    while Q is not empty:

        current = dequeue Q

        print current



        for each neighbor of current:

            if neighbor is not visited:

                mark neighbor as visited

                enqueue neighbor into Q
```

6.When BFS Is Used

✔ Finding the **shortest path**
✔ Level-by-level exploration
✔ Social media “friend suggestions”
✔ Solving puzzles (like Rubik’s cube shortest moves)
✔ Maps / GPS navigation

7. Benefits of learning BFS :

Learning BFS helps you find the **shortest path** in graphs or grids.
It is widely used in **real-life applications** like maps, social networks, and routing.
BFS strengthens your **problem-solving skills** by teaching level-wise thinking.
It is very useful in **competitive programming** and coding interviews.
BFS also builds the foundation for advanced algorithms like **Dijkstra and A***.