Breadth First Search (BFS) in Python - Viva Preparation

# 1. BFS with Queue (using deque)

## How it Works:

This version of BFS uses a queue (`deque`) to explore the graph level by level. The queue ensures that nodes are visited in the same order they were discovered (FIFO order). The algorithm starts with the chosen start node, visits it, and then gradually explores its neighbours, then their neighbours, and so on.

## Why we use this approach:

- Queue (`deque`) allows fast insertion and deletion from both ends.  
- We ensure nodes are processed in the correct BFS order.  
- The `visited` set prevents visiting the same node multiple times, avoiding infinite loops.

# 2. BFS without Queue (using current\_level and next\_level sets)

## How it Works:

This version of BFS replaces the queue with two sets: `current\_level` and `next\_level`. `current\_level` contains all nodes currently being visited, while `next\_level` collects neighbours for the next round. Once a level is processed, `current\_level` is updated to `next\_level`.

## Why we use this approach:

- This is an alternative implementation that does not need a queue.  
- It is useful to clearly separate nodes by their 'level' or 'distance' from the start node.  
- The `visited` set again ensures that no node is visited more than once.

# 3. Common Viva Questions & Answers

Q1: What is BFS?  
A: BFS is a graph traversal algorithm that visits nodes level by level, starting from a given start node.

Q2: Why do we need the `visited` set?  
A: To make sure no node is visited more than once, which prevents cycles or infinite loops.

Q3: Why is a queue (`deque`) used in BFS with queue?  
A: Because BFS requires First-In-First-Out order, and deque makes insertion/removal efficient.

Q4: How is BFS without queue different?  
A: Instead of a queue, it uses two sets (`current\_level` and `next\_level`) to process nodes level by level.

Q5: Which version is better?  
A: Both are correct. The queue version is more standard, while the level-set version is useful for problems that require knowing the exact level of each node.