

BREADTH-FIRST TRAVERSAL

Level Order



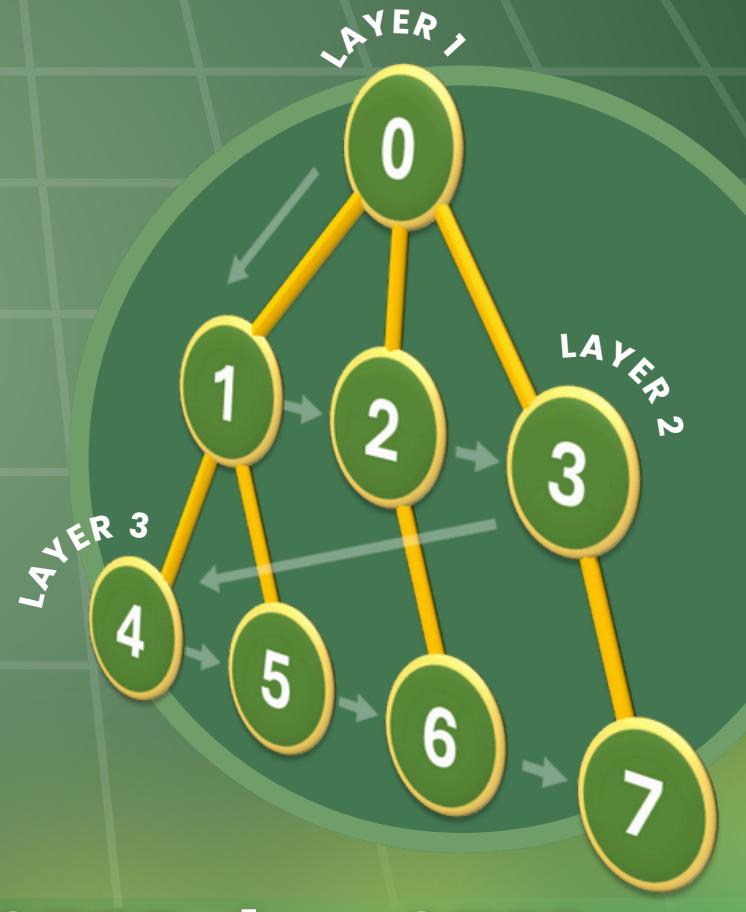
Definition

- Also known as Breadth-First Search (BFS), is an algorithm for exploring graph or tree data structures by visiting nodes layer by layer, starting from a given node.
- It uses a queue to keep track of unvisited nodes, visiting all nodes at the current depth before moving to the next level.



How it works?

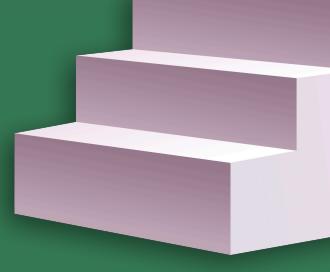
- Start at the root node.
- Use a queue to keep track of nodes to visit.
- Visit (process/print) the front node of the queue.
- Enqueue the left and right children of that node.
- Repeat steps 3–4 until the queue is empty.



STEP-by-STEP PROCESS

Refer to the diagram above

- Start with queue = [0]
- Visit 0 → enqueue 1, 2, 3 → [1, 2, 3]
- Visit 1 → enqueue 4, 5 → [2, 3, 4, 5]
- Visit 2 → enqueue 6 → [3, 4, 5, 6]
- Visit 3 → enqueue 7 → [4, 5, 6, 7]
- Visit 4 → [5, 6, 7]
- Visit 5 → [6, 7]
- Visit 6 → [7]
- Visit 7 → []



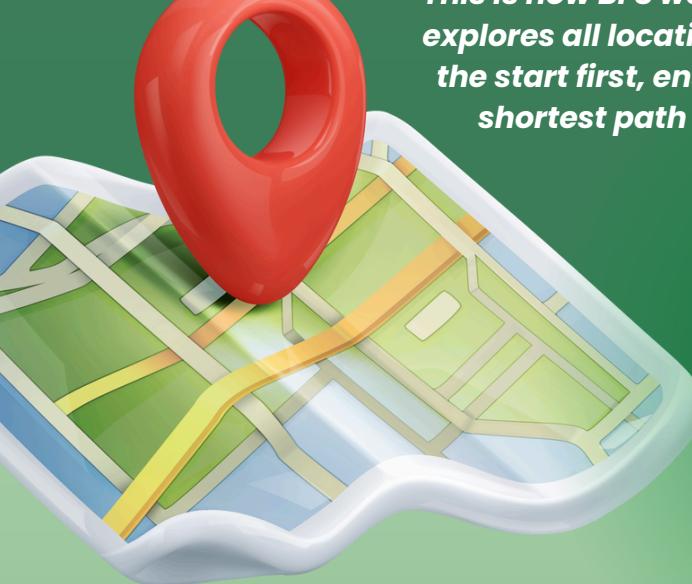
Output (Level Order):
0, 1, 2, 3, 4, 5, 6, 7

Real-Life Example

Imagine you are trying to find the shortest route to a friend's house in a city. You start from your home and:

- First, check all nearby streets (level 1).
- Then, check all streets connected to those (level 2).
- Continue outward level by level until you reach your friend's house.

This is how BFS works – it explores all locations closest to the start first, ensuring the shortest path is found.



Comparison with Depth-First Traversal

- Visits nodes by going as deep as possible before backtracking.
- Uses a stack (LIFO) or recursion.
- Only good for: exploring complete paths or checking connectivity.

Code Snippet

```
// Function for Breadth-First Traversal
void breadthFirstTraversal(int tree[8][3], int start) {
    queue<int> q;
    q.push(start);

    cout << "Breadth-First Traversal: ";

    while (!q.empty()) {
        int node = q.front();
        q.pop();
        cout << node << " ";

        // Enqueue children if they exist
        for (int i = 0; i < 3; i++) {
            if (tree[node][i] != -1)
                q.push(tree[node][i]);
        }
    }
    cout << endl;
}
```



References:

Breadth-First Search in 4



3:59

- Breadth-first search in 4 minutes** by: Michael Sambol
<https://www.youtube.com/watch?v=HZ5YTanv5QE>
- GeeksforGeeks. (2025, August 28). Breadth first search or BFS for a graph. GeeksforGeeks.
<https://www.geeksforgeeks.org/ds-a/breadth-first-search-or-bfs-for-a-graph/>