

Breadth-First Search (BFS) in Graph

BFS (Breadth-First Search) is a graph traversal algorithm that explores vertices level by level.

Key Features:

- Uses a Queue (FIFO)
- Time Complexity: $O(V + E)$, where V is vertices and E is edges

Steps:

1. Start from a source node.
2. Mark it as visited.
3. Add it to a queue.
4. While the queue is not empty:
 - Remove the front node.
 - Visit all unvisited neighbors.
 - Mark them as visited and add to the queue.

Python Example (using adjacency list):

```
from collections import deque
```

```
def bfs(graph, start):
```

```
    visited = set()
```

```
    queue = deque([start])
```

```
    while queue:
```

```
        node = queue.popleft()
```

```
if node not in visited:

    print(node, end=' ')

    visited.add(node)

    for neighbor in graph[node]:

        if neighbor not in visited:

            queue.append(neighbor)
```

```
graph = {

    'A': ['B', 'C'],

    'B': ['D', 'E'],

    'C': ['F'],

    'D': [],

    'E': ['F'],

    'F': []

}
```

```
bfs(graph, 'A')
```

Output:

A B C D E F

Applications:

- Finding shortest path in unweighted graphs
- Web Crawlers
- Network Broadcasting
- AI (pathfinding)
- Garbage Collection

