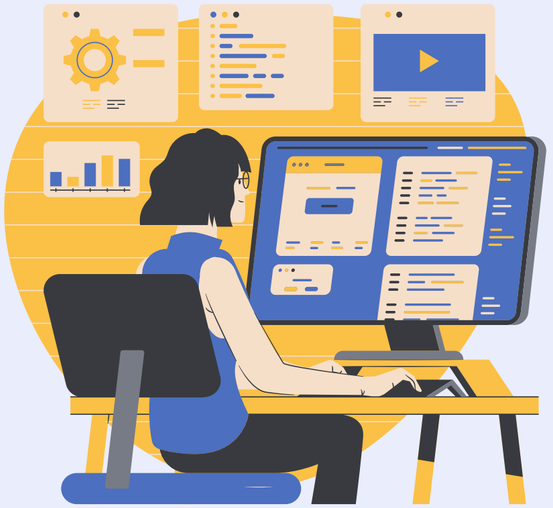
BFS



KSKLN

KSKLN

BOLATAY SHAPAGAT

BADARUDDIN ZEBUNNISA

KESSIKBAYEV MARLEN

SABITOVA KAMILA

ASHIRKHAN OLZHAS

What is BFS Algorithm?

Breadth-first search (BFS) is an algorithm that is used to graph data or searching tree or traversing structures.

The algorithm efficiently visits and marks all the key nodes in a graph in an accurate breadthwise fashion. This algorithm selects a single node in a graph and then visits all the nodes adjacent to the selected node.

Once the algorithm visits and marks the starting node, then it moves towards the nearest unvisited nodes and analyses them. Once visited, all nodes are marked. These iterations continue until all the nodes of the graph have been successfully visited and marked.

Why do we need BFS Algorithm?

There are numerous reasons to utilize the BFS Algorithm to use as searching for your dataset.

Some of the most vital aspects that make this algorithm your first choice are:

BFS is useful for analyzing the nodes in a graph and constructing the shortest path of traversing through these.

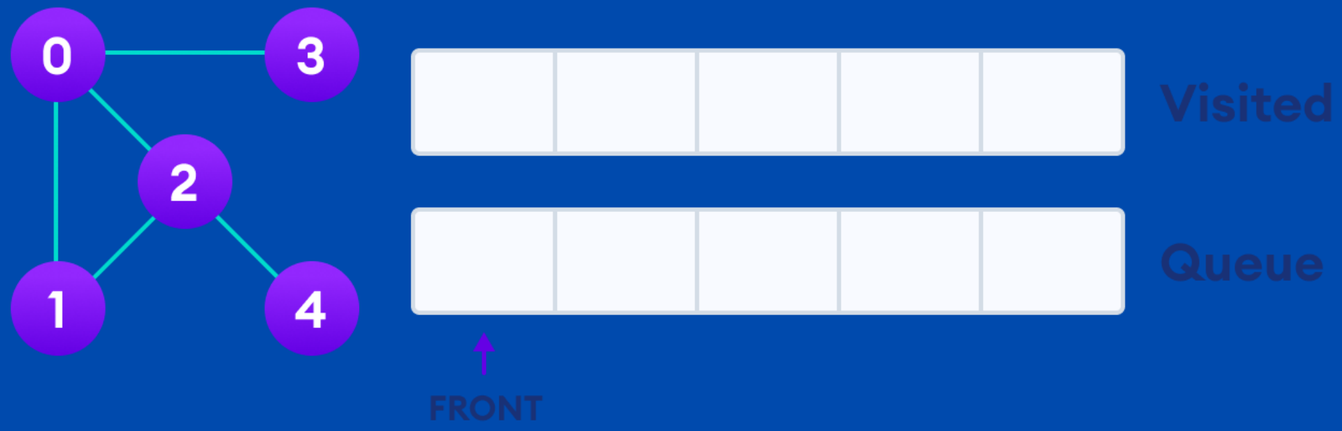
BFS can traverse through a graph in the smallest number of iterations.

BFS iterations are seamless, and there is no possibility of this algorithm getting caught up in an infinite loop problem.

The architecture of the BFS algorithm is simple and robust.

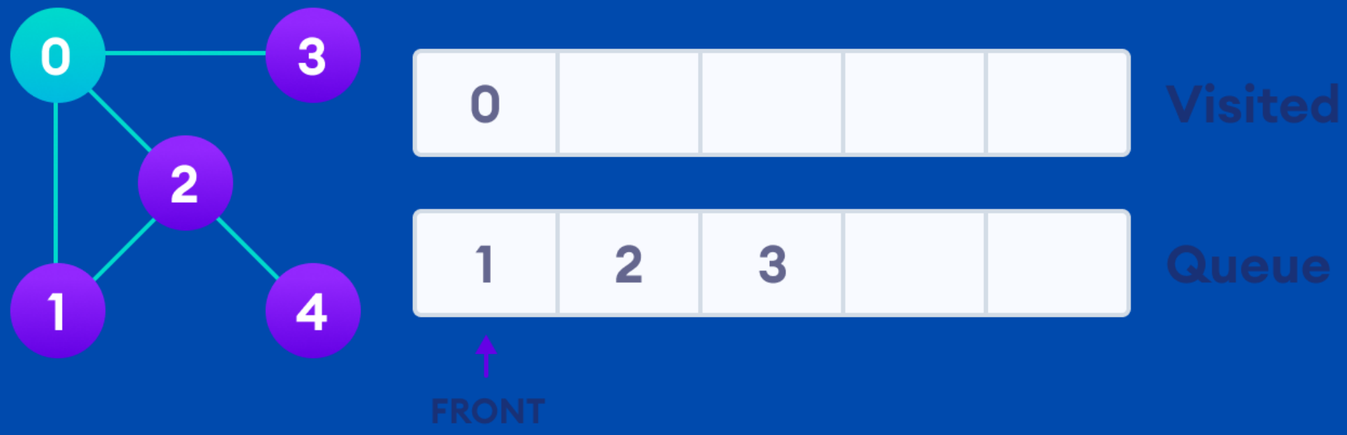
The result of the BFS algorithm holds a high level of accuracy in comparison to other algorithms.

BFS example



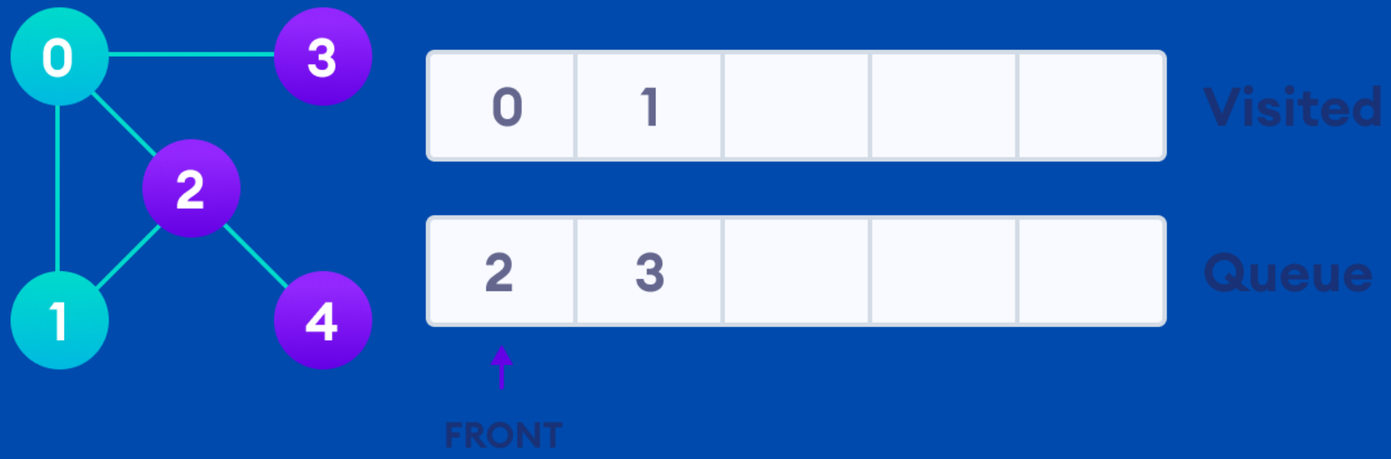
Let's see how the Breadth First Search algorithm works with an example. We use an undirected graph with 5 vertices.

Step 1



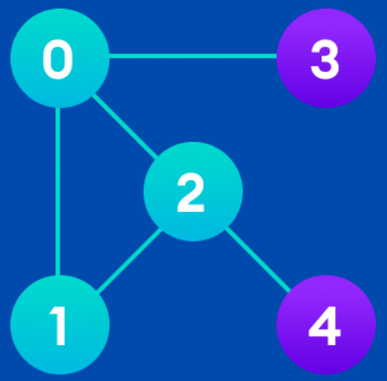
We start from vertex 0, the BFS algorithm starts by putting it in the Visited list and putting all its adjacent vertices in the stack.

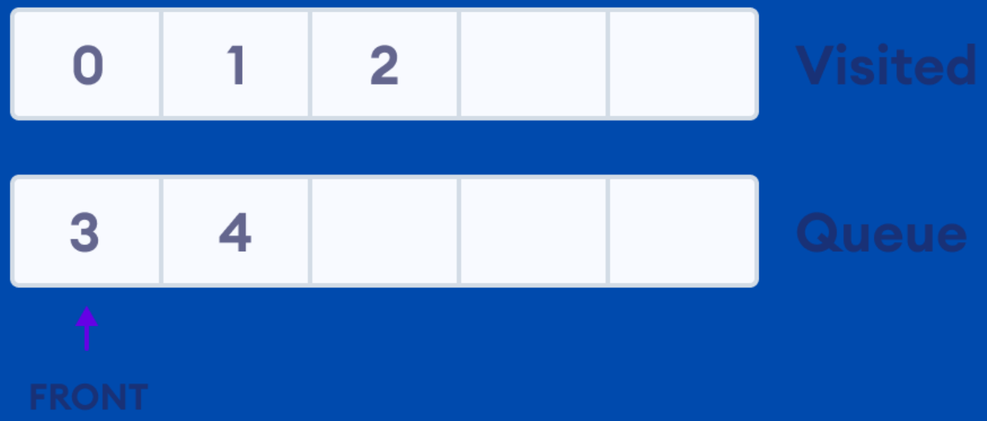
Step 2



Next, we visit the element at the front of queue i.e. 1 and go to its adjacent nodes. Since 0 has already been visited, we visit 2 instead.

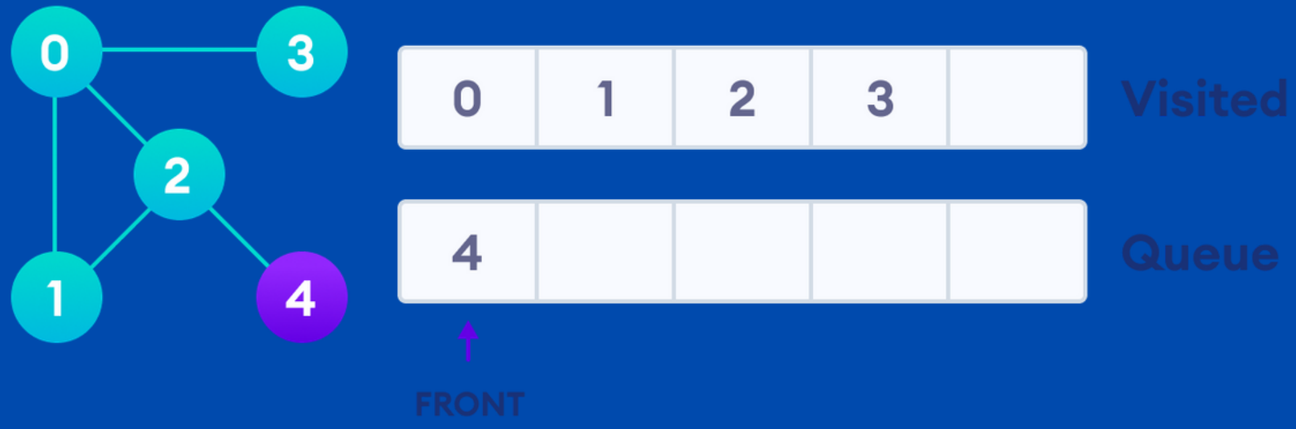
Step 3





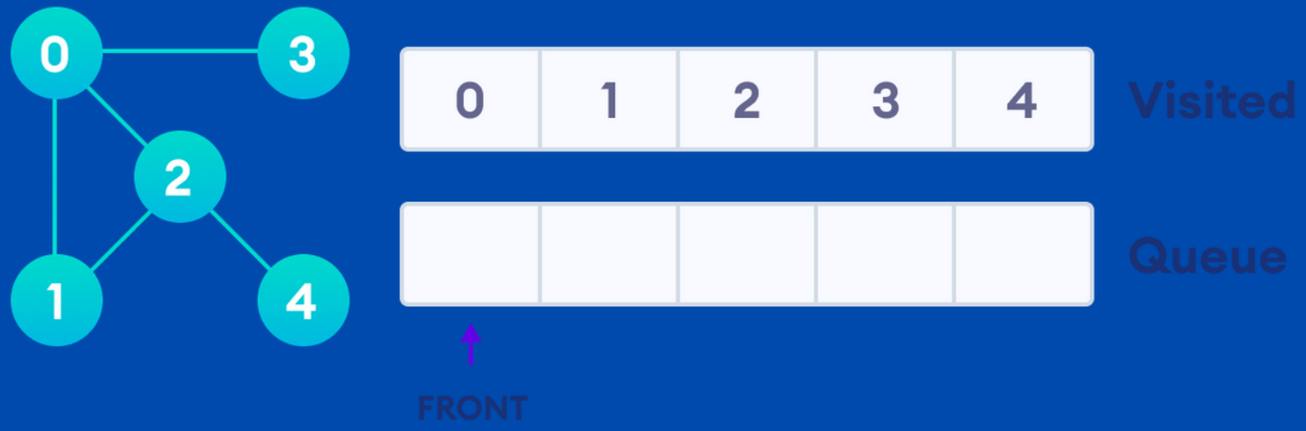
Vertex 2 has an unvisited adjacent vertex in 4, so we add that to the back of the queue and visit 3, which is at the front of the queue.

Step 4



Only 4 remains in the queue since the only adjacent node of 3 i.e. 0 is already visited. We visit it.

Step 5



Since the queue is empty, we have completed the Breadth First Traversal of the graph.

Rules of BFS Algorithm

A queue (FIFO-First in First Out) data structure is used by BFS.

You mark any node in the graph as root and start traversing the data from it.

BFS traverses all the nodes in the graph and keeps dropping them as completed.

BFS visits an adjacent unvisited node, marks it as done, and inserts it into a queue.

Rules of BFS Algorithm

Removes the previous vertex from the queue in case no adjacent vertex is found.

BFS algorithm iterates until all the vertices in the graph are successfully traversed and marked as completed.

There are no loops caused by BFS during the traversing of data from any node.

Summary

- A graph traversal is a unique process that requires the algorithm to visit, check, and/or update every single un-visited node in a tree-like structure. BFS algorithm works on a similar principle.
- The algorithm is useful for analyzing the nodes in a graph and constructing the shortest path of traversing through these.
- The algorithm traverses the graph in the smallest number of iterations and the shortest possible time.
- BFS selects a single node (initial or source point) in a graph and then visits all the nodes adjacent to the selected node. BFS accesses these nodes one by one.
- BFS selects a single node (initial or source point) in a graph and then visits all the nodes adjacent to the selected node. BFS accesses these nodes one by one.

Bye)

Thank u, for attention