

# **Breadth First Search**

## **Definition**

A breadth first search is a recursive algorithm for searching all vertices of a graph or tree data structure.

The vertices of the graph are categorized between:

1. Visited
2. Unvisited

The point of BFS is to mark each vertex as visited while avoiding cycles.

## **Pseudocode**

```
create a queue Q
function bfs(v):
    visited[v] = true
    push v into Q
    while Q is not empty do:
        u = q.pop()
        if u is not visited do:
            visited[u] = true
            for node in neighbors(u) do:
                q.push(node)
```

## **Algorithm Description**

1. Start by pushing an arbitrary vertex into the queue.
2. Pop from the front of the queue and add it to the visited list.
3. Create a list of the vertex's adjacent nodes. Add the unvisited to the queue.
4. Repeat steps 2 and 3 until the queue is empty.

## **Time Complexity**

The worst time complexity is  $O(V + E)$ .