BREADTH FIRST SEARCH

Breadth-first search (**BFS**) is an algorithm for searching a tree data structure for a node that satisfies a given property. It starts at the tree root and explores all nodes at the present depth prior to moving on to the nodes at the next depth level. Extra memory, usually a queue, is needed to keep track of the child nodes that were encountered but not yet explored.

Now let's take a look at the steps involved in traversing a graph by using Breadth-First Search:

Step 1: Take an Empty Queue.

Step 2: Select a starting node (visiting a node) and insert it into the Queue.

Step 3: Provided that the Queue is not empty, extract the node from the Queue and insert its child nodes (exploring a node) into the Queue.

Step 4: Print the extracted node.

Pseudocode :

Input : A graph *G* and a *starting vertex root* of *G*

Output : Goal state. The *parent* links trace the shortest path back to *root*

```
1   procedure BFS(G, root) is
2   let Q be a queue
3   label root as explored
4   Q.enqueue(root)
5   while Q is not empty do
6   v := Q.dequeue()
7   if v is the goal then
8   return v
9   for all edges from v to w in G.adjacentEdges(v) do
10  if w is not labeled as explored then
11  label w as explored
12  w.parent := v
13  Q.enqueue(w)
```

Time and space complexity :

The Time complexity can be expressed as O(|V|+|E|).

The Space complexity can be expressed as O(|V|).