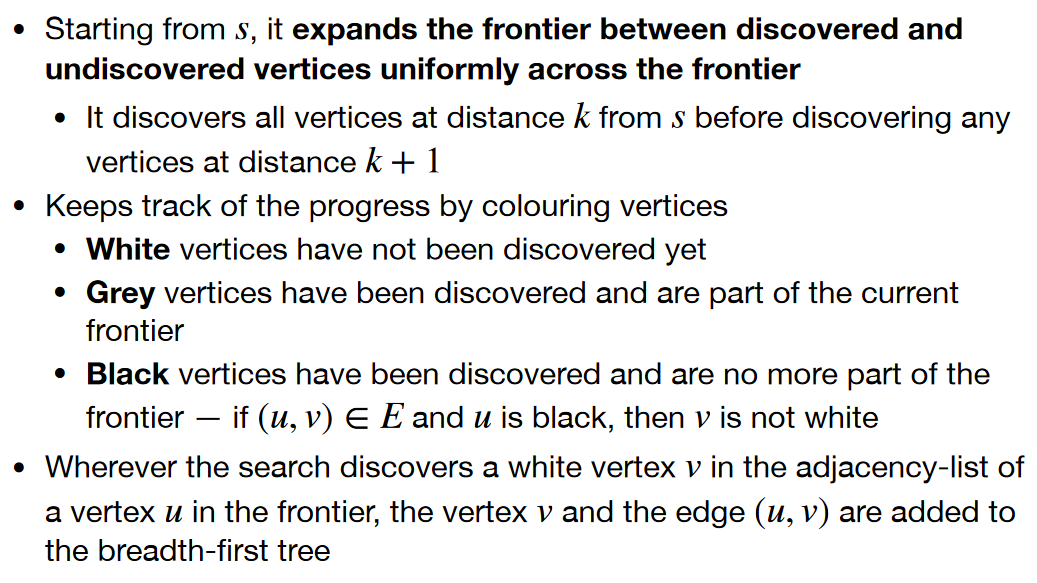
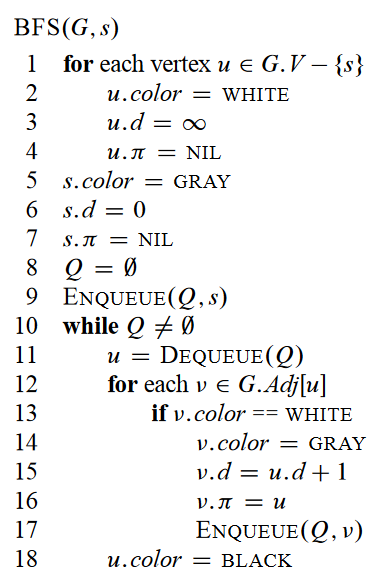
# Breadth-first search(BFS)

BFS finds the shortest path, aka the minimal amount of edges to travel to reach a point in a graph.



We have the graph *G = (V, E)* and a starting vertex *s*.

On lines 1 - 4 all vertices are initialized. The colors are explained on page 1.

*u.d =* the length from *s* to that vertex and *u.π* is the previous vertex.

On lines 5 – 7 the start vertex is initialized.

On lines 8 – 9 the array *Q*  is initialized and the start vertex *s* is *Enqueued*.

The while-loop continues until there is no more vertices. First on line 11 it removes the head of *Q* with *Dequeue* and sets *u* equal it*.*

Then the for-loop on line 12 goes for all adjacent vertices to *u.* If it’s a new vertex(if its white) it updates it color, the length from s to the vertex(*d*) and the previous vertex. Afterwards it enqueues the vertex.

If the vertex is grey, which means its already visited its color is set to black on line 18.

## Example

