Breadth-first search[1]

Procedure BFS(G, v_0)
INPUT: digraph $G = (V, E)$ in adjacency array representation
root (source) $v_0 \in V$
OUTPUT: parent links, distances from root, accessibility from v_0

```
Initialization
for v ∈ V do status(v) := WHITE, p(v) := NIL, dist(v) := ∞
Q := ∅                                   (: empty FIFO queue created :)
status(v0) := GRAY, enqueue(Q, v0)       (: root discovered, added to Q :)
dist(v0) := 0, p(v0) := v0

Main loop
while Q ≠ ∅
    u := dequeue(Q)                      (: remove head of Q, call it u :)
    for w ∈ Adj[u]                       (: exploring edge u → w :)
        if status(w) = WHITE then        (: w discovered :)
            status(w) := GRAY, p(w) := u, dist(w) := dist(u) + 1
            enqueue(Q, w)                (: w added at the tail of the queue :)
    end(for)
    status(u) := BLACK                   (: u finished :)
end(while)
return arrays status, p, dist
```

Exercises.

(1) Loop invariant: $(\forall v \in V)(v \in Q \Leftrightarrow \text{status}(v) = \text{GRAY})$
(2) In the OUTPUT, all vertices accessible from v_0 are BLACK and all vertices that are not accessible from v_0 are WHITE
(3) In the OUTPUT, $(\forall u \in V)(\text{dist}(u)$ is the distance from v_0 to $u)$
(4) A shortest path from v_0 to an accessible vertex u found by reversing the sequence $u \to p(u) \to p^2(u) \to \cdots \to p^d(u) = v_0$ where $d = \text{dist}(u)$
(5) Complexity of main loop in the unit cost model: $O(n_0 + m_0)$ where n_0 is the number of vertices accessible from v_0 and m_0 is the number of edges accessible from v_0

[1] L. B., 01-26-2021. Last updated 01-31-2025