Breadth-First Search (BFS)

In this strategy we explore the neighbors of a node before going deeper. We use the same adjacency structure as in DFS.

The time stamping is a little different. A vertex's neighbors get one more in time stamp than the vertex itself.

\[
\text{for } i := 1 \text{ to } n \text{ do } V[i].d := -1 \text{ end for;}
\]
\[
V[s].d := 0; \quad V[s].\pi := \text{nil}; \quad Q := \{s\};
\]
\[
\text{BFS} (s);
\]

BFS uses a queue \( Q \) which is initialized to a vertex \( s \) from which search begins.
Procedure BFS(A);
  while Q ≠ ∅ do
    j := DEQUE; t := V[j].adj;
    while t ≠ nil do
      if V[t.v].d = -1 then
        V[t.v].d := V[j].d + 1
        V[t.v].π := j
        ENQUEUE (t.v)
      endif
      t := t.next;
    endwhile
  endwhile

$T(n,m) = O(n+m)$
$S; a, b, c, d, e$

The BFS tree can be redrawn which represents the part of the graph reachable from $S$.

For $u \in V$ define $\delta(S,u)$ as the number of edges of a shortest path from $S$ to $u$.

Claim. If $u$ is reachable from $S$ then $V[u].d = \delta(S,u)$ after completion of BFS.