`Breitensuche` (engl. **B**readth **F**irst **S**earch)

Eingabe: Graph $G = (V, E)$; Startknoten $s \in V$.

Ausgabe: Felder $d[v], pred[v]$.

$Q := New(Queue)$;

$$d[v] := \begin{cases} 0 & \text{falls } v = s \\ \infty & \text{sonst} \end{cases}$$

Füge s in Q ein;

while $Q \neq \emptyset$ **do**

    Entferne vordersten Knoten v von Q;

    for all $w \in \Gamma(v)$ **do**

        if $d[w] = \infty$ **then**

            $d[w] := d[v] + 1$;

            $pred[w] := v$;

            Füge w am Ende von Q an;

        end if

    end do

end do

Bestimme Zusammenhangskomponenten

Eingabe: Graph $G = (V, E)$;

Ausgabe: Feld $comp[v]$ (Nummern der Zshgs.komponenten)

$comp[v] := 0$ für alle $v \in V$;

$Q := New(Queue)$; $i := 0$;

repeat

Wähle beliebigen Knoten $s \in V$ mit $comp[s] = 0$;

$i := i + 1$; $comp[s] := i$;

Füge s in Q ein;

while $Q \neq \emptyset$ **do**

Entferne vordersten Knoten v von Q;

for all $w \in \Gamma(v)$ **do**

if $comp[w] = 0$ **then**

$comp[w] := comp[v]$;

Füge w am Ende von Q an;

end if

end do

end do

until $comp[v] > 0$ für alle $v \in V$;

`Tiefensuche` (engl. **D**epth **F**irst **S**earch)

Eingabe: Graph $G = (V, E)$; Startknoten $s \in V$.

Ausgabe: Felder $dfs[v]$.

$S := New(Stack);$

$$dfs[v] := \begin{cases} 0 & \text{falls } v = s \\ \infty & \text{sonst} \end{cases}$$

$v := s; i := 0; allscanned := false;$

repeat

if $\exists w \in \Gamma(v)$ mit $dfs[w] = \infty$ **then**

$Push(S, v);$

$i := i + 1; dfs[w] := i;$

$v := w$

else if $S \neq \emptyset$ **then**

$v := Pop(S);$

else

$allscanned := true;$

end if

until $allscanned;$