

Algorithmus DFS-2ZK

Eingabe: Zusammenhängender Graph $G = (V, E)$ durch Inzidenzlisten L .

Ausgabe: Die zweifachen ZHK von G .

```
procedure suche ( $v$ : Knoten) ;
begin
  markiere  $v$  BESUCHT;
  tief[ $v$ ] := dfnr[ $v$ ] := zähler;
  zähler := zähler + 1;
  for jeden Knoten  $w$  aus  $L[v]$  do
    if  $w$  ist nicht BESUCHT
      then  $T := T \cup \{(v \rightarrow w)\}$ ; vater[ $w$ ] :=  $v$ ;
        lege  $w$  auf den Keller BESICHTIGTE_KNOTEN;
        lege  $(v \rightarrow w)$  auf den Keller BESICHTIGTE_KANTEN;
        suche( $w$ );
        if tief[ $w$ ]  $\geq$  dfnr[ $v$ ] and  $v \neq r$  (* Test der Bedingung, die  $v$  als AP nachweist *)
          then (* Ausgabe der 2fachen Zshgskomp., die  $r$  nicht enthält *)
            gib alle Knoten oben vom Keller BESICHTIGTE_KNOTEN bis  $w$  einschl. aus;
            gib  $v$  aus; (*  $v$  liegt noch immer auf dem Keller *)
            gib alle Kanten oben vom Keller BESICHTIGTE_KANTEN bis einschl.  $(v \rightarrow w)$  aus;
            schreibe „Ende einer 2fachen Zusammenhangskomponente“
          fi;
          tief[ $v$ ] :=  $\min\{\text{tief}[v], \text{tief}[w]\}$ 
        else if  $w \neq \text{vater}[v]$  and dfnr[ $v$ ]  $>$  dfnr[ $w$ ]
          then lege  $(v \rightarrow w)$  auf BESICHTIGTE_KANTEN;
            tief[ $v$ ] :=  $\min\{\text{tief}[v], \text{dfnr}[w]\}$ 
          fi
        fi
  end
begin (* Hauptprogramm *)
   $T := \emptyset$ ; zähler := 1;
  for alle  $v \in V$  do markiere  $v$  als nicht BESUCHT;
  wähle Startknoten  $r$ ;
  suche( $r$ );
  while Keller BESICHTIGTE_KANTEN nicht leer
    (* Alle 2fachen Zshgskomp., die  $r$  enthalten, sind noch nicht ausgegeben worden *)
  do
    gib alle Kanten oben auf BESICHTIGTE_KANTEN bis zur ersten Kante  $(r \rightarrow v) \in T$  einschl. aus;
    gib alle Knoten von BESICHTIGTE_KNOTEN bis  $v$  einschl. aus;
    gib  $r$  aus;
    schreibe „Ende 2fache Zusammenhangskomponente, die  $r$  enthält“
  done
end (* Hauptprogramm *)
```