

Algorithm initResult(G)  
 $S$  is new Sequence

Algorithm preComponentVisit(G, v)  
 $S.insertLast(v)$

Algorithm result(G)  
 return  $S$

Algorithm BFS(G)  
 initResult(G)

for all  $u \in G.vertices()$  do  
 setLabel( $u$ , UNEXPLORED)  
 postInitVertex( $u$ )

for all  $e \in G.edges()$  do  
 setLabel( $e$ , UNEXPLORED)  
 postInitEdges( $e$ )

for all  $v \in G.vertices()$  do  
 if getLabel( $v$ ) is UNEXPLORED  
     preComponentVisit( $G, v$ )  
     BFSComponent( $G, v$ )  
     postComponentVisit( $G, v$ )

return result(G)

Algorithm 13.55 Components( $G, s$ )

$L \leftarrow$  new list

$L.insertLast(s)$

startVertexVisit( $G, v$ )

while  $L$  is empty() do

$v \leftarrow L.remove(L.first())$

preVertexVisit( $G, v, e, w$ )

for all  $e \in G.incidentEdges(v)$  do

preEdgesVisit( $G, v, e, w$ )

if  $getLabel(e) = UNEXPLORED$  then

$w \leftarrow G.opposite(v, e)$

unexploredEdgesVisit( $G, v, e, w$ )

if  $getLabel(w) = UNEXPLORED$  ~~(false)~~ then

preDiscover

preDiscoverVisit( $G, v, e, w$ )

setLabel( $e, Discovery$ )

setLabel( $w, Visited$ )

Requeue( $w$ )

postDiscoverVisit( $G, v, e, w$ )

else

setLabel( $e, REVISIT$ )

edgesVisit( $G, v, e, w$ )

postVertexVisit( $G, v$ )

FinishVertexVisit( $G, v$ )

## Find Cycle

Algorithm InitResult( $G, v$ )  
cycle  $\leftarrow$  ~~new~~ lib

---

Algorithm result( $G$ )  
return cycle

---

Algorithm CrossEdgeVisit( $G, v, e, w$ )  
cycle.insertFirst( $v$ )  
cycle.insertLast( $e$ )  
while  $w \neq v$  do  
     $p_v \leftarrow G.parent(v)$   
     $p_e \leftarrow G.parent(e)$   
    if  $p_v \neq p_e$  then  
        cycle.insertFirst( $p_v$ )  
        cycle.insertLast( $p_e$ )  
    else  
        cycle.insertFirst( $p_v$ )