

Template Version of BFS

Algorithm **BFS**(*G*) {top level}

Input graph *G*

Output labeling of the edges of
G as discovery edges and
cross edges

initResult(*G*)

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

postInitVertices(*u*)

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

postInitEdges(*e*)

for all $v \in G.vertices()$ do
 if **getLabel**(*v*) = UNEXPLORED
 then

preComponentVisit(*G*, *v*)

BFScomponent(*G*, *v*)

postComponentVisit(*G*, *v*)

return **result**(*G*)

Algorithm **BFScomponent**(*G*, *s*)

startBFS(*G*, *s*)

setLabel(*s*, VISITED)

L ← new List

L.insertLast(*s*)

while $\neg L.isEmpty()$ do

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

preVertexVisit(*G*, *v*)

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

preEdgeVisit(*G*, *v*, *e*)

 if **getLabel**(*e*) = UNEXPLORED

$w \leftarrow opposite(v, e)$

 if **getLabel**(*w*) = UNEXPLORED

preDiscEdgeVisit(*G*, *v*, *e*, *w*)

setLabel(*e*, DISCOVERY)

setLabel(*w*, VISITED)

L.insertLast(*w*)

postDiscEdgeVisit(*G*, *v*, *e*, *w*)

 else

setLabel(*e*, CROSS)

crossEdgeVisit(*G*, *v*, *e*, *w*)

postVertexVisit(*G*, *v*)

finishBFS(*G*, *s*)