After labeling each vertex with its component number (HW13)

```
Algorithm DFS(G)
  Input graph G
  Output the edges of G are labeled as
         discovery edges and back edges

    initResult(G)
    for all u ∈ G.vertices()
        setLabel(u, UNEXPLORED)
        postVertexInit(G, u)
    for all e ∈ G.edges()
        setLabel(e, UNEXPLORED)
        postEdgeInit(G, e)
    for all v ∈ G.vertices()
        if getLabel(v) = UNEXPLORED
            preComponentVisit(G, v)
            DFScomponent(G, v)
            postComponentVisit(G, v)

    result(G)
```

```
Algorithm DFScomponent(G, v)
    setLabel(v, VISITED)
    startVertexVisit(G, v)
    for all e ∈ G.incidentEdges(v)
        preEdgeVisit(G, v, e, w)
        if getLabel(e) = UNEXPLORED
            w ← opposite(v,e)
            edgeVisit(G, v, e, w)
            if getLabel(w) = UNEXPLORED
                setLabel(e, DISCOVERY)
                preDiscoveryVisit(G, v, e, w)
                DFScomponent(G, w)
                postDiscoveryVisit(G, v, e, w)
            else
                setLabel(e, BACK)
                backEdgeVisit(G, v, e, w)
    finishVertexVisit(G, v)
```

HW14: Insert into T the smallest-weight edge going out from each component