Template Version of BFS

```
Algorithm BFS(G) {all components}
  Input graph G
  Output labeling of the edges of G as
    discovery edges and cross edges

    initResult(G)
    for all u ∈ G.vertices() do
        setLabel(u, UNEXPLORED)
        postInitVertex(u)
    for all e ∈ G.edges() do
        setLabel(e, UNEXPLORED)
        postInitEdge(e)
    for all v ∈ G.vertices() do
        if isNextComponent(G, v)
            preComponentVisit(G, v)
            BFScomponent(G, v)
            postComponentVisit(G, v)

    return result(G)

Algorithm isNextComponent(G, v)
    return getLabel(v) = UNEXPLORED
```

```
Algorithm BFScomponent(G, s) {1 component}
    setLabel(s, VISITED)
    Q ← new empty Queue
    Q.enqueue(s)
    startBFScomponent(G, s)
    while ! Q.isEmpty() do
        v ← Q.dequeue()
        preVertexVisit(G, v)
        for all e ∈ G.incidentEdges(v) do
            preEdgeVisit(G, v, e)
            if getLabel(e) = UNEXPLORED
                w ← opposite(v, e)
                edgeVisit(G, v, e, w)
                if getLabel(w) = UNEXPLORED
                    preDiscEdgeVisit(G, v, e, w)
                    setLabel(e, DISCOVERY)
                    setLabel(w, VISITED)
                    Q.enqueue(w)
                    postDiscEdgeVisit(G, v, e, w)
                else
                    setLabel(e, CROSS)
                    crossEdgeVisit(G, v, e, w)
        postVertexVisit(G, v)

    finishBFScomponent(G, s)
```