DFS for undirected graphs

```
dfs(graph, node) {
  if ("finished") return
  state[node] ← "discovered"
  time ← time + 1
  processVertexEarly(node)
  foreach neighbour ∈ Neighbourhood(node) {
    if (state[neighbour] ≠ "discovered") {
      parent[neighbour] ← node
      processEdge(node, neighbour)
      dfs(graph, neighbour)
    } else if (state[neighbour] ≠ "processed") {
      processEdge(node, neighbour)
    }
    if ("finished") return
  }
  processVertexLate(currentNode)
  state[currentNode] ← "processed"
  time ← time + 1
}
```

DFS for directed graphs

```
dfs(graph, node) {
  if ("finished") return
  state[node] $\leftarrow$ "discovered"
  time $\leftarrow$ time + 1
  processVertexEarly(node)
  foreach neighbour $\in$ Neighbourhood(node) {
    if (state[neighbour] $\neq$ "discovered") {
      parent[neighbour] $\leftarrow$ node
      processEdge(node, neighbour)
      dfs(graph, neighbour)
    } else if (state[neighbour] $\neq$ "processed" $\vee$ graph is directed) {
      processEdge(node, neighbour)
    }
    if ("finished") return
  }
  processVertexLate(currentNode)
  state[currentNode] $\leftarrow$ "processed"
  time $\leftarrow$ time + 1
}
```