```
dfs(graph, node) {
  state ← Array[n, "undiscovered"]
  finished ← false
  dfs_recur(graph, node)
}

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

node=2 neighbour=3 time=2