Pseudocode

1. DFS($G$) // main routine
2. for each vertex $u \in V$
3.     do color[$u$] = "white"
4. endfor
5. time = 0
6. for each vertex $u \in V$ do // commonly processed in alphabetical order
7.     if color[$u$] = "white" then
8.         DFS-visit($u$)
9.     endif
10. endfor // end of main routine

---

1. DFS-visit($u$) // subroutine
2. color[$u$] = "gray"
3. time = time + 1
4. $d[u] = time$ // discovery timestamp
5. for each $v \in \text{Adj}[u]$ do // commonly processed in alphabetical order
6.     if color[$v$] = "white" then
7.         DFS-visit($v$)
8.     color[$u$] = "black"
9.     time = time + 1
10. $f[u] = time$ // finish timestamp

Classification of edges

- Tree edge (T): encounter new (white) vertex (gray to white)
- Back edge (B): from descendant to ancestor (gray to gray)
- Forward edge (F): from ancestor to descendant (gray to black)
- Cross edge (C): any other edges (between trees/subtrees) (gray to black)

Remarks

- In an undirected graph, there may be some ambiguity since edge $(u, v)$ and $(v, u)$ are the same edge. Classify by the first type above that matches.
- Therefore, if $G$ is undirected, a DFS produces only tree (T) and back (B) edges.