Pseudocode

```
DFS(G)        // main routine
for each vertex u ∈ V
   do color[u] = "white"
endfor
time = 0
for each vertex u ∈ V do        // commonly processed in alphabetical order
   if color[u] = "white" then
      DFS-VISIT(u)
   endif
endfor        // end of main routine
```

—— ◇ ——

```
DFS-VISIT(u)        // subroutine
color[u] = "gray"
time = time + 1
d[u] = time        // discovery timestamp
for each v ∈ Adj[u] do        // commonly processed in alphabetical order
   if color[v] = "white" then
      DFS-VISIT(v)
color[u] = "black"
time = time + 1
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.*