Depth-First Search

DFS has a neat recursive implementation which eliminates the need to explicitly use a stack.

Discovery and final times are sometimes a convenience to maintain.

```
DFS(G)
for each vertex $u \in V[G]$ do
        $color[u] = white$
        $parent[u] = nil$
$time = 0$
for each vertex $u \in V[G]$ do
        if $color[u] = white$ then DFS-VISIT[u]
```

Initialize each vertex in the main routine, then do a search from each connected component. BFS must also start from a vertex in each component to completely visit the graph.

```
DFS-VISIT[u]
$color[u] = grey$ (*u had been white/undiscovered*)
$discover[u] = time$
$time = time + 1$
for each $v \in Adj[u]$ do
        if $color[v] = white$ then
                $parent[v] = u$
                DFS-VISIT(v)
$color[u] = black$ (*now finished with $u$*)
$finish[u] = time$
$time = time + 1$
```