0.1 Depth-first Search

Algorithm 1 DFS

Let G be a graph with vertex set $V = \{1, ..., n\}$

```
procedure DFS($G, V$)
    label $v$ as discovered
    for all edges from $v$ to $w$ in $G$.adjacentEdges($V$) do
        if (vertex w is not labelled as discovered) then
            recursively call DFS($G, w$)
        end if
    end for
end procedure
```

This algorithm allows you to find the connected components of a disconnected graph. Then using the following algorithm we can check if our forest at each step of our algorithm is acyclic.

Algorithm 2 Acyclic Check

Let G be a graph with the set of connected componenets C as found by DFS(G,v) where v is an arbitrary vertex in G.

```
procedure ACYCLIC($G, C$)
    for all $i$ in $C$ do
        if $i$.edgeCount() $> n - 1$ then return False
        end if
    end forreturn True
end procedure
```