Correctness:

Claim: After explore(v) has terminated: For every vertex u,

visited(u) = true $\Leftrightarrow$ $\exists$ path from v to u.

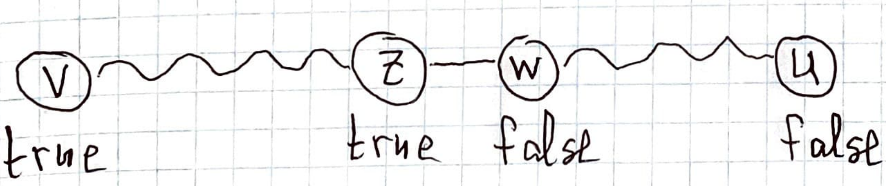
Proof: $\Rightarrow$ follows from the algorithm: the algorithm "walks" from a vertex to a neighboring vertex.

$\Leftarrow$ Assume $\exists$ path from v to u.

Consider an arbitrary path from v to u.

Assume at termination, visited(u) = false.

Let z be the last vertex on this path for which visited(z) = true at termination.
Let w be the vertex on this path after z.
Note: At termination, visited(w) = false



At the moment when visited(z) is set of true, the call explore(z) generates the call explore(w), during which visited(w) is set to true. $\lightning$

QED

Undirected graph $G = (V, E)$.

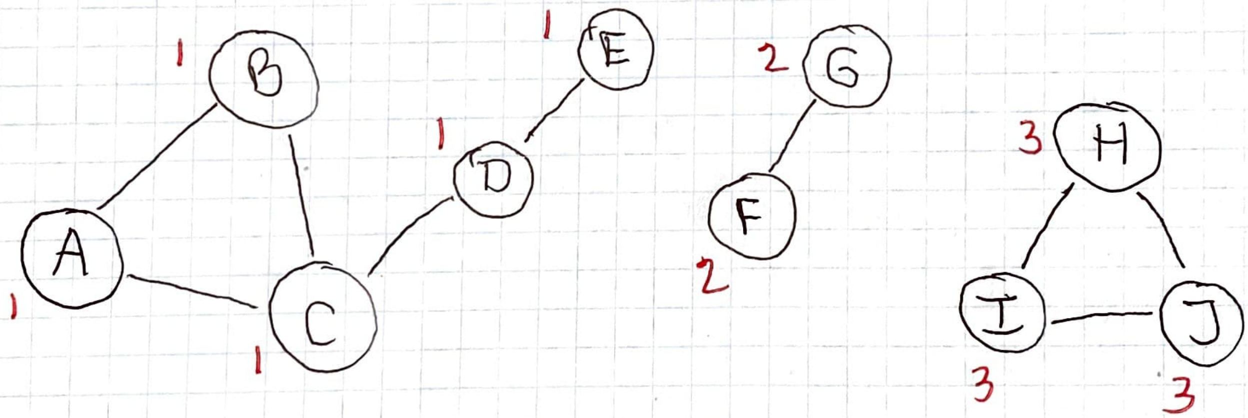
How to compute the connected components of G?

At termination:

number the connected components as $1, 2, 3, \ldots$

for each vertex v:

ccnumber(v) = number of the connected component that vertex v belongs to.



Algorithm DFS (G) : // depth-first search

for all v ∈ V : visited (v) = false ;

cc = 0 ;

for all v ∈ V :

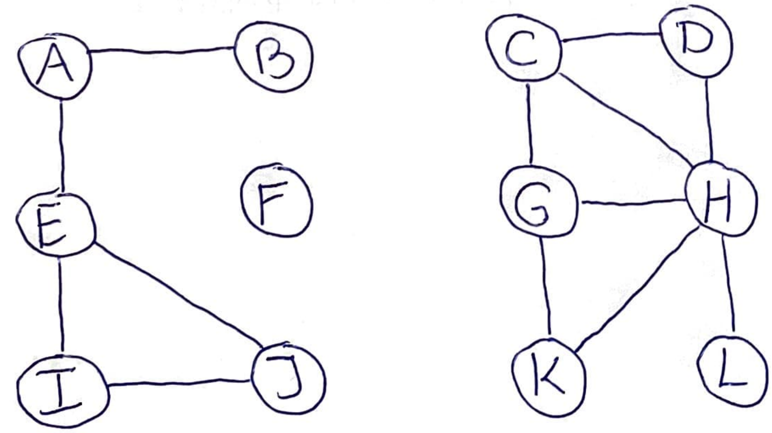
    if visited (v) = false : cc = cc + 1 ;
                                        explore (v)

In algorithm explore(v) : (see page 68)

previsit (v) ≡ "ccnumber (v) = cc"

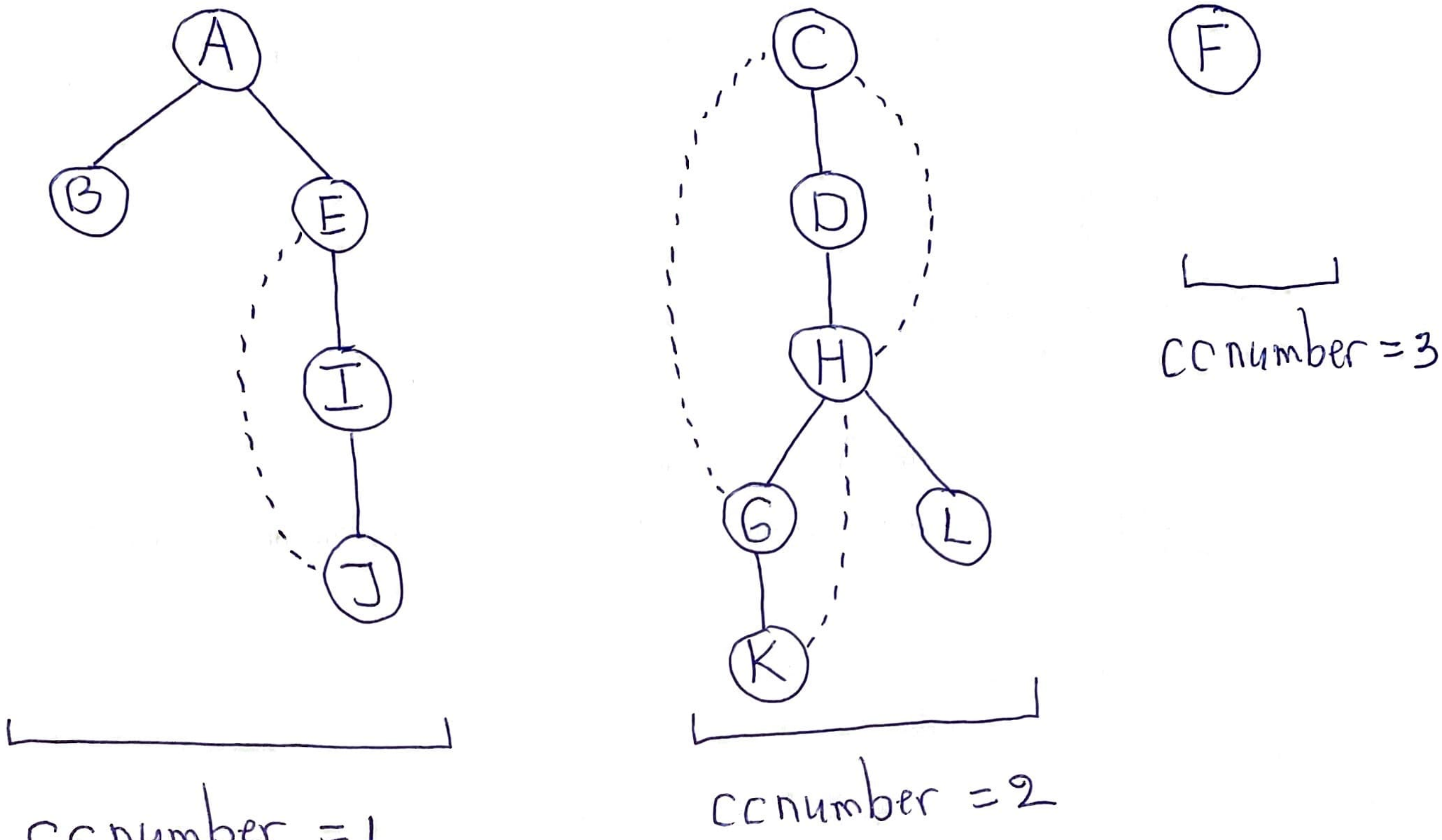
postvisit (v) ≡ "empty"

Run DFS(G) on :



assume: adjacency lists are sorted alphabetically.

The result is the following DFS-forest:



Running time of DFS:

* first for-loop: $O(|V|)$
* second for-loop:
  - explore(u) is called exactly once for each vertex u (this may be part of a recursive call)
  - time spent for explore(u), excluding recursive calls, is $O(1 + \text{degree}(u))$.

Total time:

$$O\left(|V| + \sum_{u \in V} (1 + \text{degree}(u))\right)$$

$$= O(|V| + |E|)$$