

## Try it Yourself!

DFSWrapper(G)

  counter = 0

  For each vertex u of G  
    If u is not "seen"  
      DFS(u)

  End If

End For

DFS(u)

  Mark u as "seen"

  u.start = counter++

  For each edge (u,v) //leaving u

    If v is not "seen"

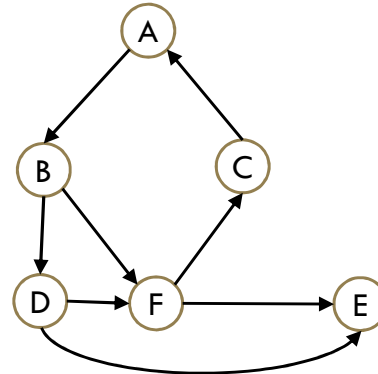
      DFS(v)

    End If

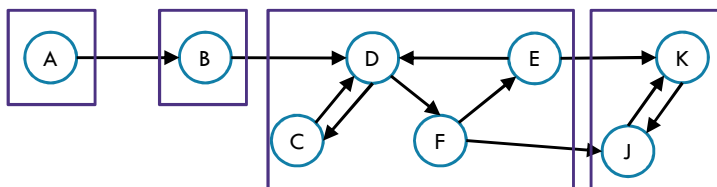
  End For

  u.end = counter++

Type	Definition	When is (u,v) that edge type?
Tree	Edges forming the DFS tree (or forest).	v was not seen before we processed (u,v).
Forward	From ancestor to descendant in tree.	u and v have been seen, and u.start < v.start < v.end < u.end
Back	From descendant to ancestor in tree.	u and v have been seen, and v.start < u.start < u.end < v.end
Cross	Edges going between vertices without an ancestor relationship.	u and v have not been seen, and v.start < v.end < u.start < u.end



## Problem 2: Find Strongly Connected Components



{A}, {B}, {C,D,E,F}, {J,K}

### Strongly Connected Component

A subgraph C such that every pair of vertices in C is connected via some walk **in both directions**, and there is no other vertex which is connected to every vertex of C in both directions.

## Graph Modeling Process

1. What are your fundamental objects?  
Those will probably become your vertices.
2. How are those objects related?  
Represent those relationships with edges.
3. How is what I'm looking for encoded in the graph?  
Do I need a path from  $s$  to  $t$ ? The shortest path from  $s$  to  $t$ ? A minimum spanning tree? Something else?
4. Do I know how to find what I'm looking for?  
Then run that algorithm/combination of algorithms  
Otherwise go back to step 1 and try again.

## Scenario #1

You've made a new social networking app, Convrs. Users on Convrs can have "asymmetric" following (I can follow you, without you following me). You decide to allow people to form multi-user direct messages, but only if people are probably in similar social circles (to avoid spamming).

You'll allow a messaging channel to form only if for every pair of users  $a, b$  in the channel:  $a$  must follow  $b$  or follow someone who follows  $b$  or follow someone who follows someone who follows  $b$ , or ...  
And the same for  $b$  to  $a$ .

You'd like to be able to quickly check for any new proposed channel whether it meets this condition.

What are the vertices?

What are the edges?

What are we looking for?

What do we run?