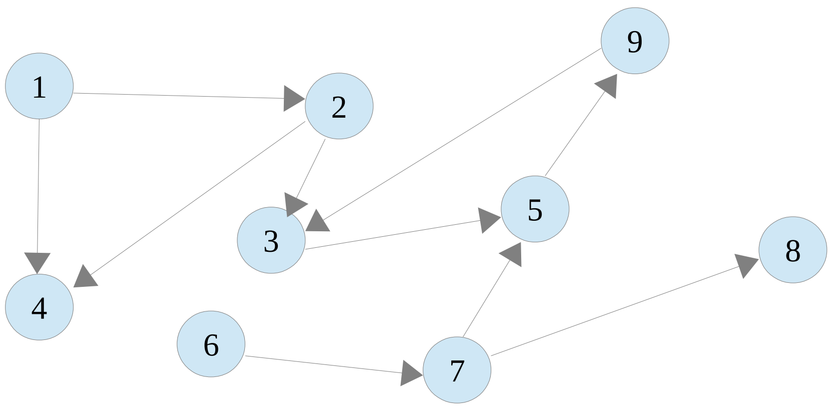
DESCHINKEL Karine november 2013

Exam on Graphs (CP58) - 1H30 -

Exercise 1 (/10)

1)Write the adjacency matrix and the incidence matrix of the following directed acyclic graph
2) Give the degree of each vertex
3) Show the (DFS) depth-first spanning tree (highlight tree edges, back edges, forward edges, cross edges with different colors). Start at vertex 1 for your DFS , and when there is a choice of which edge to follow, always choose the one to the vertex that is first numerically.
4) Apply an algorithm (which one?) to detect if there is cycle in the graph. Show the different steps of the algorithm on this example (starting from vertex 1).
5) Is there articulation point(s) in the graph ? If yes, quote it (them).
6) Imagine that this graph is a contact graph for assembly of a particular device. First change the direction of the edge (3,9). Apply an algorithm (which one?) to remove redundant constraints of precedence. Write down the simplified graph and the first step of the algorithm.
7) Give a valid assembly sequence for this contact graph.



Exercise 2 (/4)

1) Give the definition of a strongly connected component in a directed graph?
2) Suppose G=(V,E) is a directed graph. Here is an algorithm to detect strongly connected Components in a directed graph

Step 1 : Call a DFS1 (Depth First Search) for the graph G and compute the postorder PO

```
DFS1(G)
{
        PO(x) <- 0 for all x in V
        counter ← 0
        WHILE there exists a vertex s such that PO(s)=0
                choose one vertex s, mark s
                 FOR each unmarked successor w of s
                        DFSRec1(w,counter)
                ENDFOR
                counter ++
                PO(s) ← counter
        ENDWHILE
}
```

```
DFSRec1(x,counter)
{
        mark x
        FOR each unmarked successor y of x
                DFSRec1(y,counter)
        ENDFOR
        counter ++
        PO(x) ← counter

}
```

Step 2 : Consider the reverse graph G^T . The reverse graph G^T of a directed graph G is another directed graph on the same set of vertices with all of the edges reversed compared to the orientation of the corresponding edges in G. That is, if G contains an edge (u,v) then the reverse of G contains an edge (v,u) and vice versa.

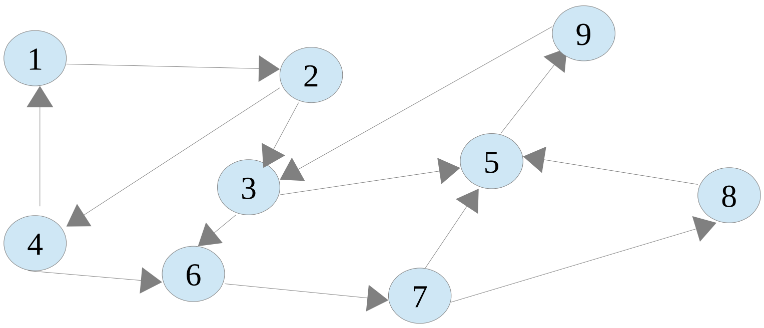
Step 3 : Call a DFS for the graph G^T, starting from vertex v with the maximal postorder,

```
DFS2(G^T)
{
        SCC(x) <- 0 for all x in V
        counter ← 0
        WHILE there exists a vertex s such that SCC(s)=0
                choose one vertex s with maximal PO, mark s
                counter ++
                SCC(s) ← counter
                FOR each unmarked successor w of s
                        DFSRec2(w,counter)
                ENDFOR

        ENDWHILE
}

DFSRec2(x,counter)
{
        mark x
        SCC(x) ← counter
        FOR each unmarked successor y of x
                DFSRec2(y,counter)
        ENDFOR
}
```

Try to apply this algorithm on this graph. Quote the different strongly connected components of the graph. SCC gives the number of strongly connected component which the vertex x belongs.



Exercise 3 (/4)

One step in the procedure for generating assembly sequences in the software ORASSE consists in identifying some sub-assemblies by searching for specific patterns in the graph. A pattern is a subgraph of the simplified contact graph of the product and contains at minimum 2 vertices of the same color (2 components assigned to the same technical function). Figure 1 shows two different patterns in the contact graph. One indicator of modularity is based on the minimal distance d between any pair of colored vertices in sub-assemblies.

Let define $M=(V_M, E_M)$ a coloured pattern.
V_C is the set of colored vertices in the pattern M.

λ_u= min { **distance**(u,v), v in V_C}, Δ=min {λ_u, u in V_C} .

Here, the **distance** between two vertices in a graph is the minimal number of edges in a shortest path connecting them (regardless of the direction of the arrows).

1. Propose an algorithm based on a known algorithm which allows to compute Δ for a particular pattern M.
2. Apply your method on the pattern on figure 2. Detail main steps of your method.

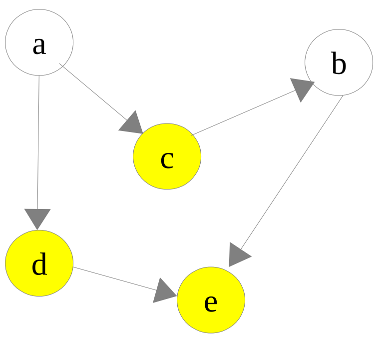


figure 1 : a pattern M with 5 components

$M=(V_M, E_M)$ a coloured pattern.
V_M = {a,b,c,d,e}
E_M = {(a,d), (a,c), (c,b), (b,e), (d,e)}
V_C is the set of colored vertices in the pattern M.

V_C = {c,d,e}

λ_u= min { **distance**(u,v), v in V_C} :

λ_c= min {distance (c,d) = 2, distance (c,e)=2}=2

λ_d= min {distance (d,c) = 2, distance (d,e)=1}=1

λ_e= min {distance (e,c) = 2, distance (e,d)=1}=1

Δ=min {λ_u, u in V_C} :

Δ=min {λ_c , λ_d, λ_e }=1

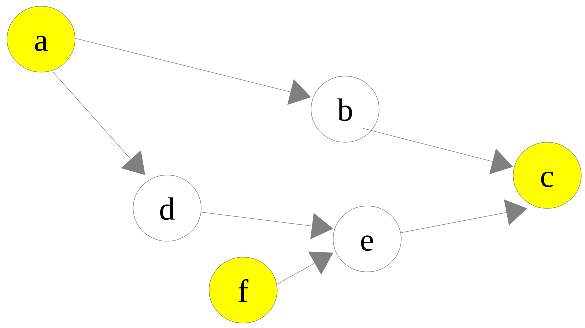


figure 2 : a pattern M with 6 components, $V_C = \{a,f,c\}$

Exercise 4 (/2)

Imagine an algorithm based on the classification of edges during the execution of a DFS to simplify a contact graph (eliminate redundant constraints). Apply your algorithm on this contact graph and give some explanation. Write down the resulting graph.