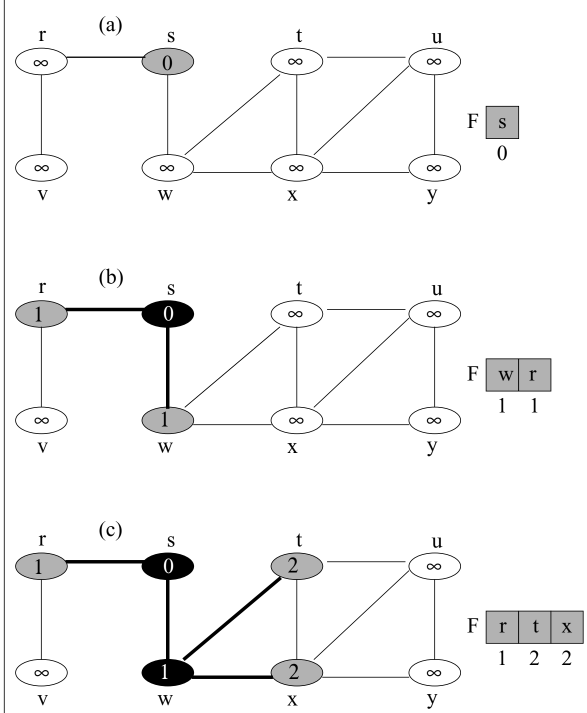
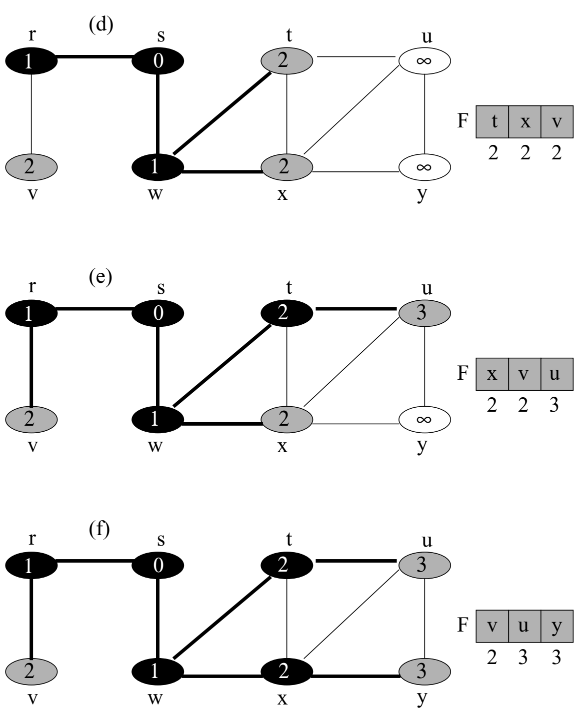
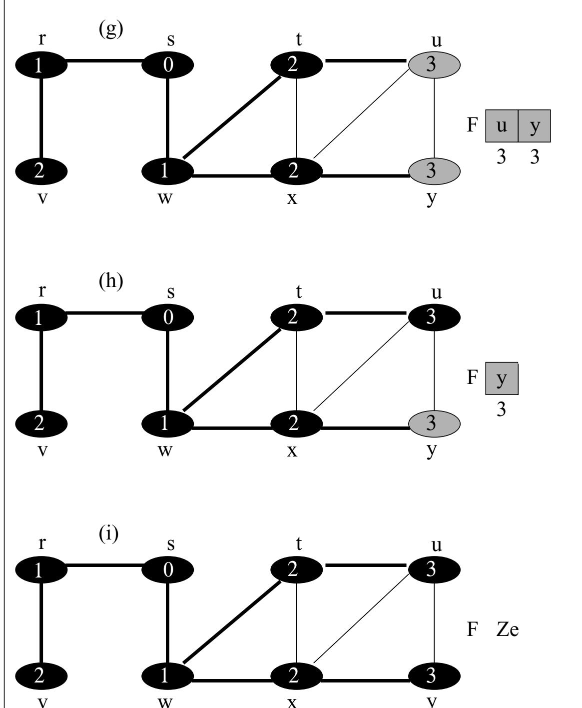
Parcours en largeur.

```
PL(G,s)
pour chaque sommet u∈ S[G]-{s}
    faire couleur[u] ← BLANC
        d[u] ← ∞
        π[u] ← NIL
couleur[s] ← GRIS
d[s] ← 0
π[s] ← NIL
F ← {s}
tant que F ≠ ∅
    faire u ← tête[F]
        pour chaque v∈ Adj[u]
            faire si couleur[v] = BLANC
                alors couleur[u] ← GRIS
                    d[v] ← d[u]+1
                    π[v] ← u
                    ENFILE(F,v)
        COULEUR[u] ← NOIR
        DÉFILE(F)
```

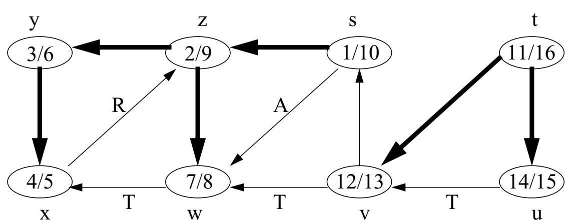


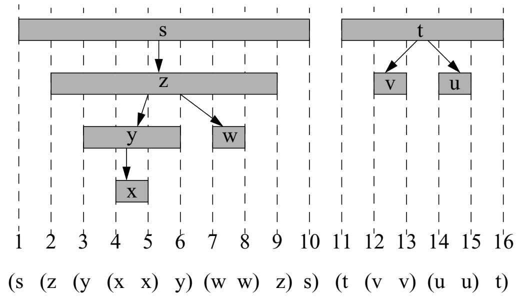


Parcours en profondeur.

PP(G)
1 **pour** chaque sommet u∈ S[G]
2 **faire** couleur[u] ← BLANC
3 π[u] ← NIL
4 date ← 0
5 **pour** chaque sommet u S ∈ [G]
6 **faire si** couleur[u] = BLANC
7 **alors** VISITER-PP(u)

VISITER-PP(u)
1 couleur[u] ← GRIS
2 date ← date+1
3 d[u] ← date
4 **pour** chaque v∈ Adj[u] // exploration de l'arc (u,v)
5 **faire si** couleur[v] = BLANC
6 **alors** π[v] ← u
7 VISITER-PP(v)
8 couleur[u] ← NOIR
9 f[u] ← date ← date+1



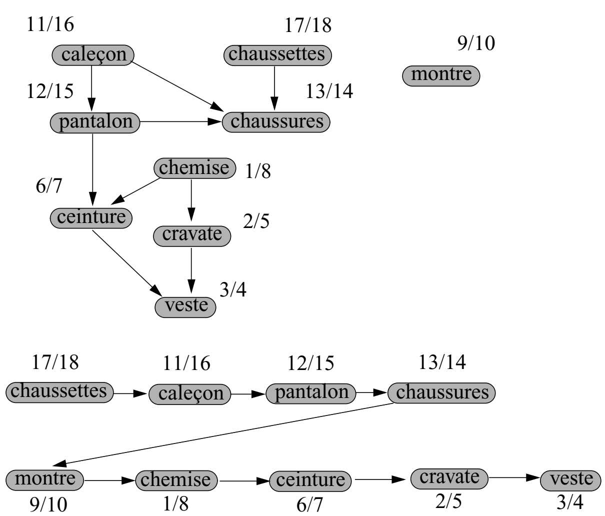


temps d'exécution

propriété de la structure des parenthèses

les quatre types d'arcs
- arcs de liaison
- arcs arrières R
- arcs avants A
- arcs transverses T

Tri topologique



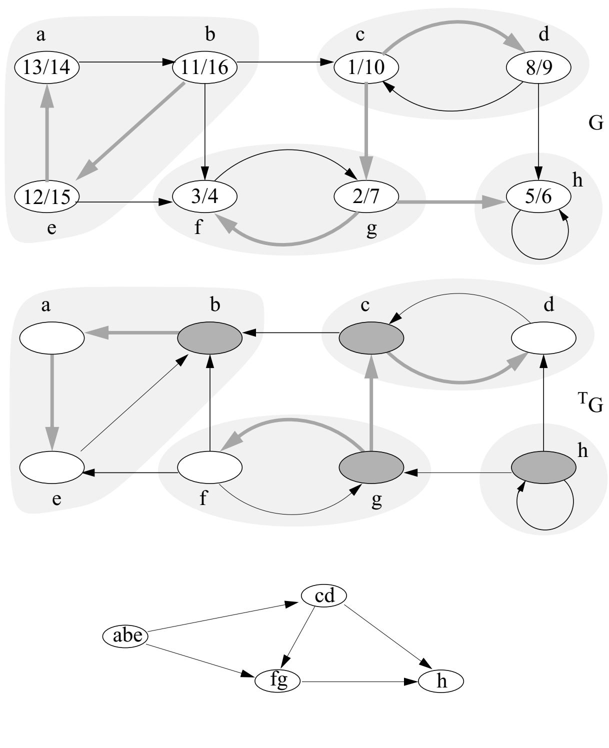
TRI-TOPOLOGIQUE(G)

1 appeler PP(G) pour calculer les dates de fin de traitement f[v] pour chaque sommet v
2 chaque fois que le traitement d'un sommet se sommet se termine, insérer le sommet début d'une liste chaînée
3 retourner la liste chaînée des sommets

Composantes fortement connexes d'un digraphe

Soit ^{T}G denote graphe G transposé (avec les arêtes inversées).

COMPOSANTES-FORTEMENT-CONNEXES(G)
1. appeler PP(G) pour calculer les dates de fin de traitement f[u] pour chaque sommet u
2. calculer ^{T}G
3. appeler PP(^{T}G), mais dans la boucle principale de PP, traiter les sommets par ordre de f[u],(calculés en ligne 1) décroissants
4. imprimer les sommets de chaque arborescence de la forêt obtenue en ligne 3 en tant que composante fortement connexe distincte



INF4393 - Jurek Czyzowicz

$f(U) = \max_{u \in U}\{f[U]\}$

Lemme.

S'il y a un arc de C à C' alors f(C) > f(C')

Autres applications du parcours en profondeur

vérifier l'acyclicité d'un graphe

vérifier l'arborescence d'un graphe