Séance de TD 09

1 Exercice 1 : Tri topologique

1.1 Introduction

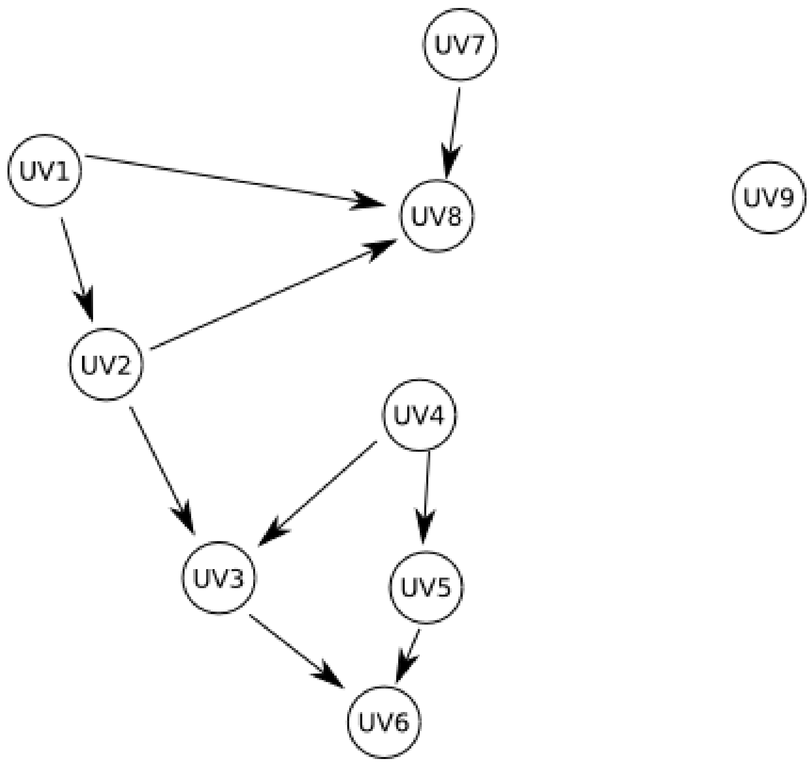
Soit $G(S, A)$ un graphe orienté sans circuit. L'algorithme de tri topologique de G consiste à ordonner linéairement tous les sommets de G tel que $\forall(u, v) \in A \times A$, u apparaît avant v dans l'ordre obtenu.

Donnez l'algorithme ainsi que son coût.

But : Comment utiliser le parcours en profondeur pour effectuer un tri topologique d'un graphe G ?
Le tri topologique consiste à :

- Ordonner linéairement tous les sommets de G.
- Enfiler certains sommets avant les autres. D'autres peuvent être traités dans n'importe quel ordre.
- $G = (S, A)$ graphe orienté sans circuit, trié topologiquement comme une suite de sommets sur une ligne horizontale de telle sorte que tous les sommets sont ordonnés de gauche à droite par ordre décroissant de date de fin de traitement.

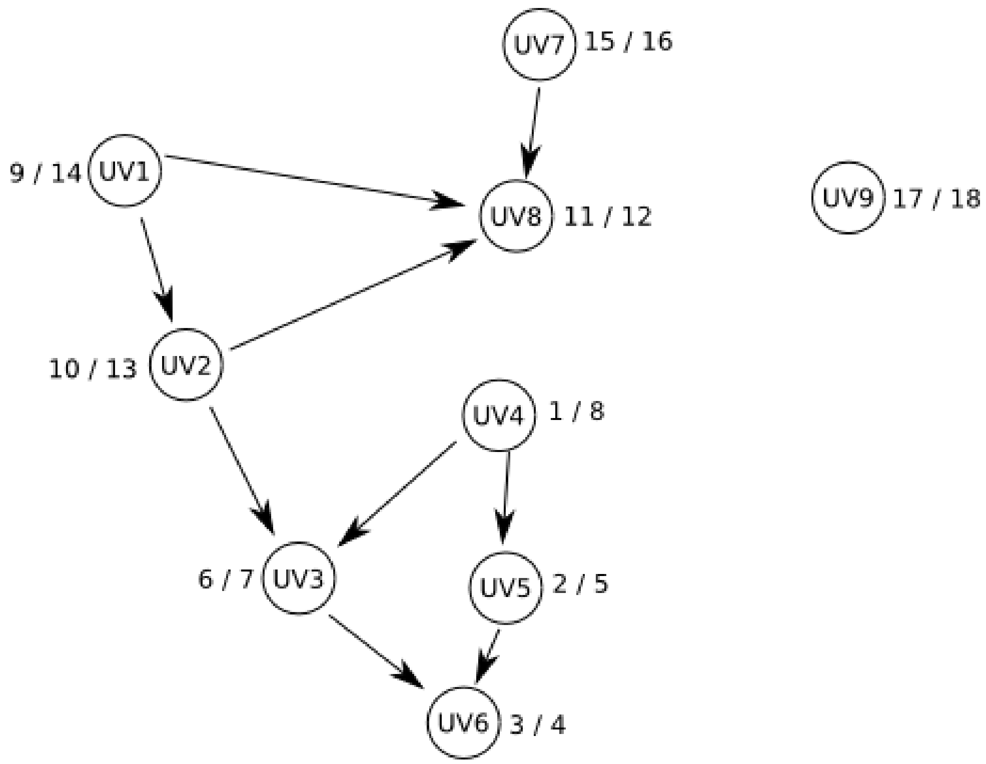
Exemple avec le système des UVs :



1.2 Algorithme du tri topologique

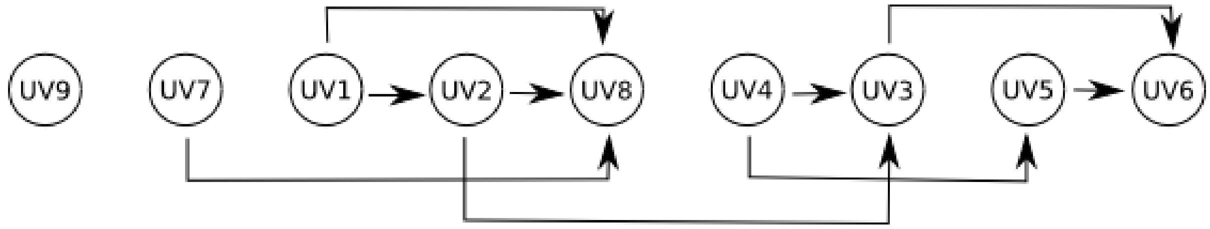
- Appeler $PP[G]$ pour calculer les dates de fin de traitement $f[v]$ pour chaque sommet V.
- Chaque fois que le traitement d'un sommet se termine, insérer le sommet au début de la liste chaînée.
- Retourner la liste chaînée.

Application sur l'exemple précédent :



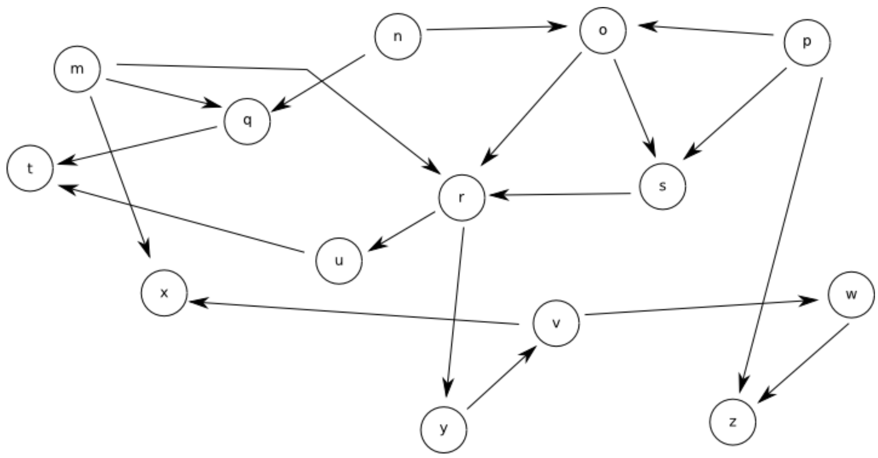
On commence arbitrairement par le sommet UV4. On descend jusqu'à UV6, puis on remonte en indiquant les dates de fin, avant de descendre vers UV3, etc.

On obtient la liste "chaînée" suivante :

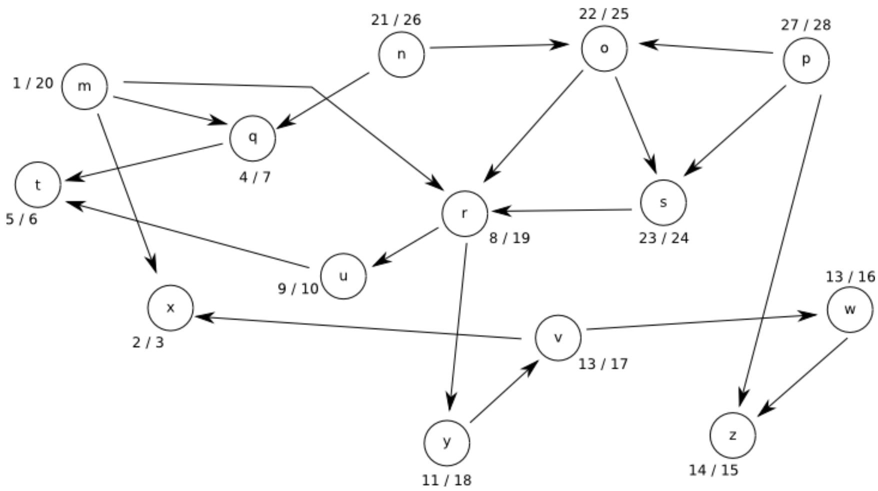


1.3 Deuxième application

Soit le graphe suivant :



L'application de l'algorithme, en commençant par le sommet m donne :



Soit la liste suivante :

p n o s m r y v w z u q t x

2 Exercice 2

Composantes fortement connexes d'un graphe.

Une CFG (composante fortement connexe) d'un graphe orienté $G = (S, A)$ est l'ensemble maximal des sommets $U \subseteq S$ tel que $\forall (u, v) \in U$, nous avons u lié à v et v lié à u (directement ou indirectement).

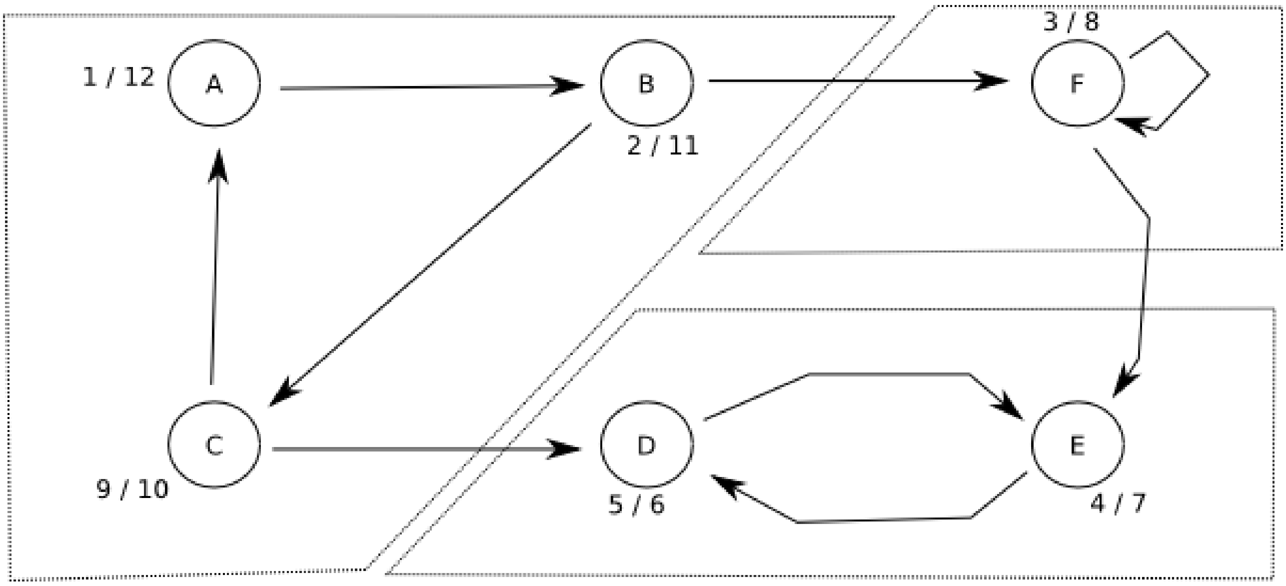
L'algorithme $\in FC$ consiste à décomposer le graphe en ses composantes fortement connexes, en utilisant le parcours en profondeur 2 fois.

Donnez cet algorithme ainsi que son coût

Ci-dessous, la description des étapes de l'algorithme :

1. $PP(G)$ (i.e. "Parcours en Profondeur").
2. Calculer T_G (transposée du graphe)
3. $PP(T_G)$
4. Afficher les sommets de chaque arborescence de la forêt obtenue à l'étape 3 en tant que composante fortement connexe distincte.

2.1 Application



On obtient la série de sommets suivante :

$$\mathbf{A} \rightarrow \mathbf{B} \rightarrow \mathbf{C}\ \mathbf{F} \rightarrow \mathbf{E} \rightarrow \mathbf{D}$$

Propriété : On remarque qu'une composante fortement connexe s'assimile à un circuit sur le graphe.