DM #5 — Exercices de graphe des vacances de printemps

Exercice 1. *Transposé d'un DAG*

On rappelle qu'un DAG est un graphe orienté sans cycle.

1. Montrez que le graphe transposé d'un DAG est un DAG.

Exercice 2. *Acyclicité du graphe des composantes fortement connexes*

On définit (un peu informellement) le *graphe des composantes fortement connexes* (GCFC) d'un graphe $G = S,A$ comme le graphe orienté $H = S',A'$ dont les sommets « représentent » les composantes fortement connexes de G (notamment $\#S'$ est égal au nombre de c.f.c. de G) et soit $v, w \in S'$ et C_v, C_w les c.f.c associées de G, $(v,w) \in A'$ ssi. $\exists v' \in C_v, \exists w' \in C_w.\ (v',w') \in A$.

1. Soit G un graphe orienté quelconque, montrez que le graphe de ses composantes fortement connexes H est un DAG (ne possède pas de cycle).

Exercice 3. *Coupe particulière dans un DAG particulier*

(Adapté d'un exercice de Jeff Erickson)

Dans cet exercice, on considère un DAG connexe $G = S,A$ représenté par tableau de listes d'adjacence et possédant un unique sommet s de degré entrant nul (c'est à dire tel qu'il n'y a aucun arc orienté $* \to s$), appelé *source*, et un unique sommet t de degré sortant nul (c'est à dire tel qu'il n'y a aucun arc orienté $t \to *$), appelé *puits*. On définit la (s,t)-*coupe* de G comme l'ensemble maximal de sommets de G distincts de s et t tel que pour tout sommet v appartenant à cette coupe, tout chemin $s \rightsquigarrow t$ passe par v (ou de façon équivalente, le sous-graphe de G induit par $S \setminus \{v\}$ ne possède pas de chemin $s \rightsquigarrow t$). *Cf.* Fig. 1.

L'objectif de cet exercice est de trouver un algorithme efficace (en temps linéaire en la taille de la représentation) pour calculer la (s,t)-coupe de G.

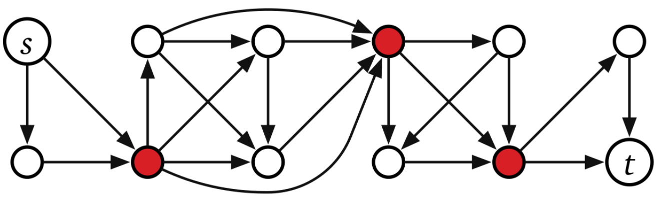

FIGURE 1 – Un exemple de *(s,t)*-coupe (en rouge). Crédit : Jeff Erickson

Deux options. Vous pouvez chercher à résoudre l'exercice sans aucune indication (c'est ainsi qu'il a été posé par son auteur original), ou suivre les étapes proposées ci-dessous.

1. Montrez qu'il existe un ordre topologique pour G tel que s et t sont respectivement minimaux et maximaux pour cet ordre. (Ceci est en fait vrai pour tous les ordres topologiques possibles, comme le montre implicitement la question suivante.)
2. Montrez que pour tout sommet v quelconque de G, on a que v est accessible depuis s ($v = s$ ou il existe un chemin $s \rightsquigarrow v$) et t est accessible depuis v.

Pour tout sommet $v \neq s, t$, on note P_v (resp. S_v) l'ensemble des sommets de G plus petits (resp. plus grands) que v pour un ordre topologique quelconque satisfaisant la condition de la première question.

3. Montrez que v appartient à la coupe ssi. il n'existe d'arc $a \to b$ pour aucun $a \in P_v$, $b \in S_v$.
4. Déduisez de ce qui précède un algorithme de coût $O(\#S\#A)$ permettant de calculer la (s,t)-coupe de G.
5. Améliorez (au besoin) votre algorithme précédent pour qu'il s'exécute en temps $O(\#A \log \#S + \#S)$, ou même $O(\#A + \#S)$.
   Indication. Une structure de donnée auxiliaire comme une file de priorité peut être utile, mais elle n'est pas nécessaire (et est à éviter pour atteindre le meilleur coût possible).

Exercice 4. *Trouver tous les isthmes* ⋆

Dans tout cet exercice, on considère un graphe non-orienté $G = S, A$ connexe (le propos se généralise immédiatement à un graphe non connexe). On appelle *isthme* de G toute arête $a \in A$ t.q. le graphe $G' := S, A \setminus \{a\}$ n'est pas connexe. On rappelle également qu'un *arbre couvrant* de G en est un sous-graphe $G'' = S, A''$ connexe.

1. Montrez que si une arête est un isthme alors elle appartient à tout arbre couvrant de G.
2. Donnez un encadrement du nombre possible ι d'isthmes pour un graphe connexe quelconque de N sommets, et un exemple de graphe atteignant chacune des bornes.
3. Montrez qu'une arête est un isthme ssi. elle n'appartient à aucun (éventuel) cycle de G.
4. Déduisez de ce qui précède un algorithme trouvant tous les isthmes de G (représenté par tableau de listes d'adjacence) en temps linéaire en la taille de sa représentation.
   Conseils. Commencez par écrire l'algorithme immédiatement suggéré par les questions précédentes sans vous soucier du coût, puis essayez de l'améliorer pour atteindre le coût cible. Pour cela, il peut être intéressant de d'abord faire le parcours, **puis** de faire ce qu'il faut par ordre de pré-visite croissant en s'arrêtant dès que possible.