Algorithmique avancée
Devoir surveillé

Les calculatrices ne sont pas autorisées.
Les exercices peuvent être traités dans le désordre.
La notation prendra en compte le soin et la clarté de la rédaction.

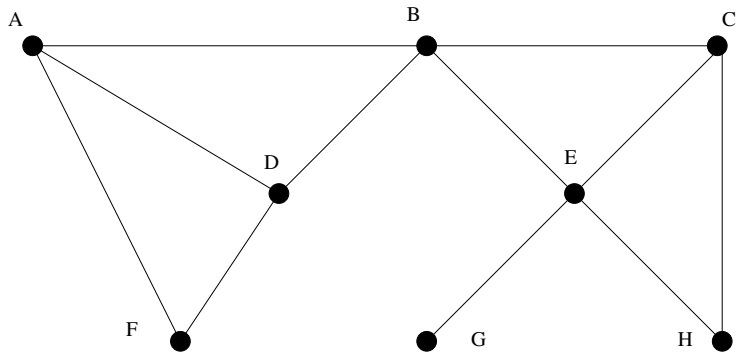

FIGURE 1 –

Exercice 1.

On considère le graphe de la figure 1.

1. Appliquer à ce graphe un arbre de parcours en profondeur de racine A. Dessinez l'arbre obtenu de haut en bas, et de gauche à droite.
2. Appliquer également un arbre de parcours en largeur de racine F. Dessinez l'arbre obtenu de haut en bas, et de gauche à droite.

Exercice 2.

Une entreprise doit poser des câbles permettant de relier différentes villes en un réseau connexe. Le prix de la pose des câbles entre deux villes est donné dans la matrice suivante, le signe ∞ désignant l'impossibilité de poser un câble directement entre ces deux villes.

$$\begin{array}{c} \\ A \\ B \\ C \\ D \\ E \end{array} \begin{array}{c} \begin{array}{ccccc} A & B & C & D & E \end{array} \\ \begin{pmatrix} 0 & 50 & 70 & \infty & 80 \\ 50 & 0 & 80 & 200 & \infty \\ 70 & 80 & 0 & 100 & 50 \\ \infty & 200 & 100 & 0 & \infty \\ 80 & \infty & 50 & \infty & 0 \end{pmatrix} \end{array}$$

1. Construire un réseau connexe le moins cher. Précisez quel algorithme est utilisé et dessinez le graphe auquel vous l'appliquez. Explicitez les étapes de l'algorithme (la solution m'importe en fait peu, elle se voit vu la taille du graphe).
2. On suppose qu'un coût est associé à l'utilisation d'une ville comme connexion, c'est-à-dire au fait qu'elle ait un degré supérieur ou égal à deux dans le réseau construit. Ce coût est de plus dépendant de la ville.

   Le coût d'un réseau est donc maintenant égal au coût des câbles plus le coût associées aux villes de degré supérieur ou égal à deux.

   Comment adapter l'algorithme existant pour résoudre le problème existant ?

   Illustrer votre proposition sur l'exemple précédent avec des coûts de 0 pour A, 20 pour B, C et 40 pour D, E.

Exercice 3.

Une *partition* d'un ensemble V est une famille d'ensembles $V_1, \ldots, V_k$ qui sont disjoints deux à deux, et dont l'union vaut V.

Une *source* d'un graphe orienté est un sommet u dont le degré entrant $d^-(u)$ est nul.

Soit G un graphe non-orienté. G est un *graphe de niveau* s'il existe une partition $V_1, \ldots, V_k$ de V telle que :

- V_1 est l'ensemble des sources de G
- Pour tout $i \geq 2$, V_i est l'ensemble des sources de $G \setminus \{V_1, \ldots, V_{i-1}\}$.

1. Les graphes de la figure 2 sont-ils des graphes de niveau ? Si oui, précisez les ensembles V_i.
2. On considère l'algorithme suivant :

   ```
   L = ∅
   Tant que G admet des sources
           Choisir une source v de G
           Ajouter v à la liste L
           Supprimer v de G
   Renvoyer L
   ```

   Appliquer cet algorithme aux deux graphes de la figure 2.

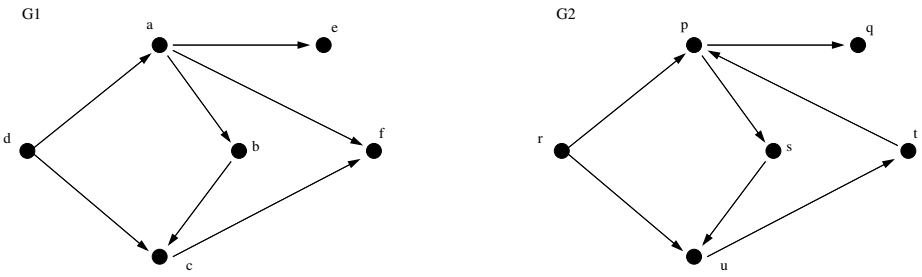

FIGURE 2 –

3. Démontrer que l'algorithme renvoie une liste contenant tous les sommets si et seulement si G est un graphe de niveau.
4. Justifier qu'un graphe de niveau ne contient pas de cycle orienté.
5. Montrer que tout graphe G sans cycle orienté a forcément une source. (On pourra raisonner par l'absurde, en supposant que tout sommet a un prédécesseur.)
6. Déduire de la question précédente que si G n'a pas de cycle, G est un graphe de niveau.
7. Ecrire un algorithme répondant à la question *Un graphe orienté G est-il acyclique ?*. Quelle est sa complexité ?

Remarque : Les graphes de niveau sont plus couramment appelés DAG, qui est l'acronyme de *Directed Acyclic Graph*.