# 9 Quelques algorithmes

Dans ce paragraphe, nous abordons quelques algorithmes appliqués à des graphes. Les algorithmes utilisés par les entreprises de réseaux sociaux sont en général bien plus complexes que ceux présentés ici mais cela vous donne une idée de l'intérêt du point de vue « graphe » lorsqu'on doit étudier un réseau.

## 9.1 Nombre d'arêtes

1. Dessiner un graphe G (non orienté) de sommets A, B, C, D, E, F dans lequel (a) A est adjacent à chacun des autres sommets. (b) B est voisin de D et E. (c) C et D sont voisins.
2. On considère la fonction ci-dessous prenant en paramètre un graphe non orienté quelconque.

```
fonction f(graphe):
    compteur ← 0
    Pour chaque sommet s du graphe:
        pour chaque voisin de s:
            ajouter 1 à compteur
    renvoyer compteur
```

   (a) Quelle est la valeur de f(G) ?

   (b) Comment peut-on modifier la fonction f pour qu'elle renvoie la taille du graphe donné en entrée ? Expliquer.

3. Taille d'un graphe complet.

**Définition.**
On appelle graphe complet un graphe (non orienté) dans lequel tout sommet est relié à tout autre.

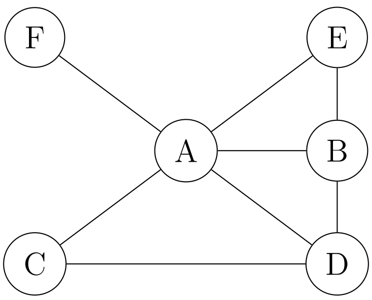
   (a) Dessiner un graphe complet d'ordre 3. Quelle est sa taille ?

   (b) Dessiner un graphe complet d'ordre 4. Quelle est sa taille ?

   (c) Quelle est la taille d'un graphe complet à $n$ sommets ?

**Résolution.**

1. Représentation du graphe :



2. (a) On déroule l'algorithme pas à pas.

    i. On commence par initialiser le compteur à 0.

    ii. On aborde la boucle «Pour chaque sommet s» :

    - Pour s = A : on exécute la boucle «pour chaque voisin de s» (ici «pour chaque voisin de A»). Cette boucle consiste à ajouter 1 au compteur pour chaque voisin de A. On ajoute donc en fait à compteur le nombre de voisins de A, c'est à dire 5. compteur a donc maintenant pour valeur 5.
    - Pour s = B : on exécute la boucle «pour chaque voisin de s» (ici «pour chaque voisin de B»). Cette boucle consiste à ajouter 1 au compteur pour chaque voisin de B. On ajoute donc en fait à compteur le nombre de voisins de B, c'est à dire 3. compteur a donc maintenant pour valeur 8.
    - Pour s = C : compteur prend la valeur 10 après ce tour de boucle.
    - Pour s = D : compteur prend la valeur 13 après ce tour de boucle.
    - Pour s = E : compteur prend la valeur 15 après ce tour de boucle.
    - Pour s = F : compteur prend la valeur 16 après ce tour de boucle.

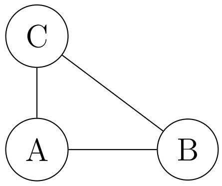
    Au final, compteur a pour valeur 16.

   (b) - La fonction a renvoyé le double du nombre d'arêtes du graphe. Et ce sera le cas pour tout graphe non orienté donné en entrée à la fonction $f$. En effet :
     — Pour chaque sommet s du graphe, on ajoute le nombre d'arêtes issues de s à compteur.
     — De cette façon chaque arête est comptée exactement deux fois. Par exemple, l'arête reliant A et B est comptée une première fois comme arête issue de A puis une seconde fois comme arête issue de B : elle est donc comptée exactement deux fois.
   - Comme f renvoie le double de la taille, il suffit de diviser par deux la valeur renvoyée pour obtenir la taille du graphe. On obtient ainsi la fonction taille ci-dessous :

```
fonction taille(graphe) :
    compteur ← 0
    Pour chaque sommet s du graphe :
        pour chaque voisin de s :
            ajouter 1 à compteur
    renvoyer compteur/2
```

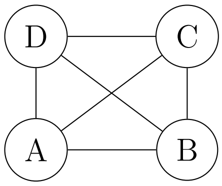
3. Taille d'un graphe complet.

   (a) Dessin d'un graphe complet d'ordre 3 :




   La taille est égale à 3.

(b) Dessin d'un graphe complet d'ordre 4 :




La taille est 6.

(c) Remarquons tout d'abord que chaque sommet est de degré $n-1$ dans un graphe complet à $n$ sommets (puisque chaque sommet est rélié à chacun des $n-1$ autres sommets).
Appliquons la fonction taille définie plus haut.
On initialise un compteur à 0.

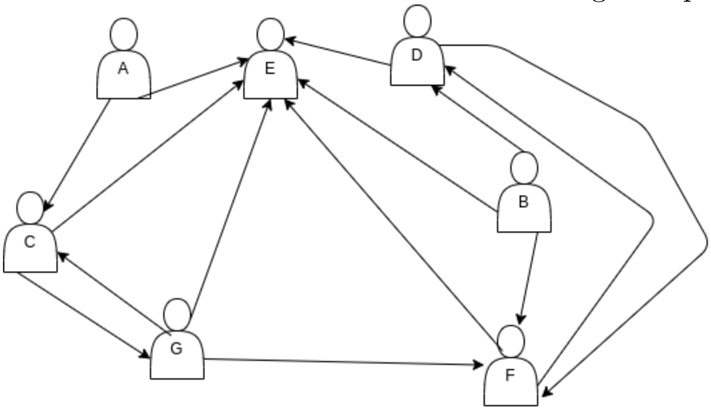
i. Pour chaque sommet s du graphe, on ajoute son nombre de voisins (c'eest à dire $n-1$) à compteur.

ii. Comme il y a $n$ sommets, le compteur vaut $n(n-1)$ à la fin de la boucle.

iii. Enfin, la taille est compteur/2, soit $\boxed{\dfrac{n(n-1)}{2}}$ □

## 9.2 La star du réseau

Dans un réseau social, on a la possibilité de « suivre » un autre abonné du réseau. Ci-dessous, une flèche d'un sommet X vers un sommet Y signifie que X suit Y.




Dans un réseau, on appelle star une personne suivie par tous mais ne suivant personne.

1. Le réseau ci-dessus présente-t-il une star ?
2. Combien de flèches faudrait-il retirer pour qu'il n'y ait plus de star dans le réseau ?
3. Peut-on ajouter des flèches de façon à ce qu'il y ait une seconde star dans le réseau ? **Justifier**.
4. Combien peut-il y avoir de stars dans un réseau ?
5. Voici un algorithme :

```
Pour chaque abonné p du réseau :
    test ← vrai
```

```
    Pour chaque abonné q différent de p:
        si p suit q alors test ← faux
        si q ne suit pas p alors test ← faux
    Si test a pour valeur vrai alors afficher p
```

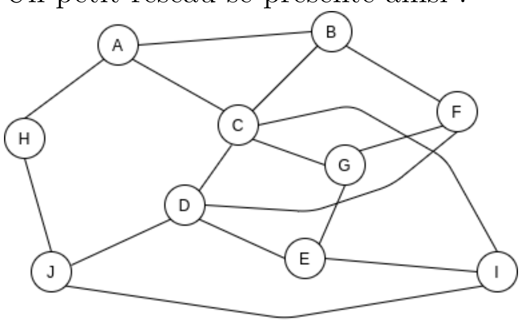
Quel sera le résultat de cet algorithme appliqué au réseau ci-dessus ? De façon générale, quel est le rôle de l'algorithme ?

**Résolution.**

1. E est une star du réseau.
2. Une seule flèche suffit. On retire une flèche arrivant sur E : ce n'est plus une star (et aucun autre sommet ne l'est).
3. Non. En effet soit Y une star. Prenons X un autre des sommets. X suit Y ( X⟶Y) puisque Y est une star, donc X suit au moins une personne et n'est pas une star. Ainsi si Y est une star, aucun autre sommet ne peut être star.
4. D'après ce qui précède, il y a au plus une star dans un réseau.
5. 
```
Pour chaque abonné p du réseau :
    test ← vrai
    Pour chaque abonné q différent de p:
        si p suit q alors test ← faux
        si q ne suit pas p alors test ← faux
    Si test a pour valeur vrai alors afficher p
```
   Cet algorithme affiche le nom du sommet star si le graphe contient une star, et n'affiche rien si le graphe ne présente pas de star.

## 9.3 Amis communs

Un petit réseau se présente ainsi :




1. Remplir le tableau suivant en appliquant l'algorithme ci-dessous. Les cases vides actuelles du tableau sont supposées contenir un 0 au départ de l'algorithme.

| | A | B | C | D | E | F | G | H | I | J |
|---|---|---|---|---|---|---|---|---|---|---|
| A | | | | | | | | | | |
| B | | | | | | | | | | |
| C | | | | | | | | | | |
| D | | | | | | | | | | |
| E | | | | | | | | | | |
| F | | | | | | | | | | |
| G | | | | | | | | | | |
| H | | | | | | | | | | |
| I | | | | | | | | | | |
| J | | | | | | | | | | |

```
Pour chaque personne i du réseau :
    Pour chaque personne j du réseau :
        Pour chaque personne k du réseau :
            si k est ami avec i :
                si k est ami avec j :
                    ajouter 1 à la case (ligne,colonne)=(i,j)
```

2. Remplir de même le tableau suivant avec l'algorithme ci-dessous :

| | A | B | C | D | E | F | G | H | I | J |
|---|---|---|---|---|---|---|---|---|---|---|
| A | | | | | | | | | | |
| B | | | | | | | | | | |
| C | | | | | | | | | | |
| D | | | | | | | | | | |
| E | | | | | | | | | | |
| F | | | | | | | | | | |
| G | | | | | | | | | | |
| H | | | | | | | | | | |
| I | | | | | | | | | | |
| J | | | | | | | | | | |

```
Pour chaque personne k du réseau :
    Pour chaque ami i de k :
        Pour chaque ami j de k :
            ajouter 1 à la case (i,j)
```

3. Vous avez normalement constaté que les deux algorithmes ont rempli la grille de la même façon.
   (a) Quelle est la signification des nombres obtenus dans les cases de la diagonale (cases A-A, B-B, ...) ?
   (b) Quelle est la signification des nombres obtenus dans les autres cases ?
   (c) L'un des deux algorithmes vous paraît-il plus rapide ? Expliquer pourquoi il demande moins de calculs.

Les progrès des ordinateurs, téléphones, etc... sont autant dus à l'amélioration des algorithmes au fur et à mesure des découvertes par les chercheurs qu'à l'amélioration matérielle. Les modifications pour augmenter la rapidité de calcul sont bien sûr en général beaucoup plus complexes que celles observées ci-dessus.

**Résolution.**

On constate que les deux algorithmes comptent, pour chaque couple (X,Y) de sommets le nombre de voisins communs.

Lorsque X=Y, ce nombre est tout simplement le nombre de voisins de X.
Le début de la matrice est donc ce qui suit pour chacun des deux algorithmes :

| | A | B | C | D | E | F | G | H | I | J |
|---|---|---|---|---|---|---|---|---|---|---|
| A | 3 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 |
| B | 1 | 3 | 1 | 2 | 0 | 0 | 2 | 1 | 1 | 0 |
| C | 1 | 1 | 5 | 0 | 3 | 3 | 0 | 1 | 0 | 2 |
| D | 1 | 2 | 0 | | | | | | | |
| E | 0 | 0 | 3 | | | | | | | |
| F | 1 | 0 | 3 | | | | | | | |
| G | 1 | 2 | 0 | | | | | | | |
| H | 0 | 1 | 1 | | | | | | | |
| I | 1 | 1 | 0 | | | | | | | |
| J | 1 | 0 | 2 | | | | | | | |

Le second algorithme demande un peu moins de calcul, car il ne considère pas tous les sommets du graphe dans les trois boucles : il ne considère que les voisins du sommet de la boucle la plus externe dans les deux boucles internes. □