À LA DÉCOUVERTE DES GRAPHES ET DES ALGORITHMES DE GRAPHES

Christian Laforest

edp sciences

À LA DÉCOUVERTE DES GRAPHES ET DES ALGORITHMES DE GRAPHES

Christian Laforest

edp sciences

Imprimé en France

ISBN : 978-2-7598-1830-3

Table des matières

Résumé visuel des principales notions utilisées dans ce livre.

Un graphe = des sommets et des arêtes.

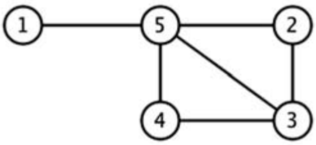


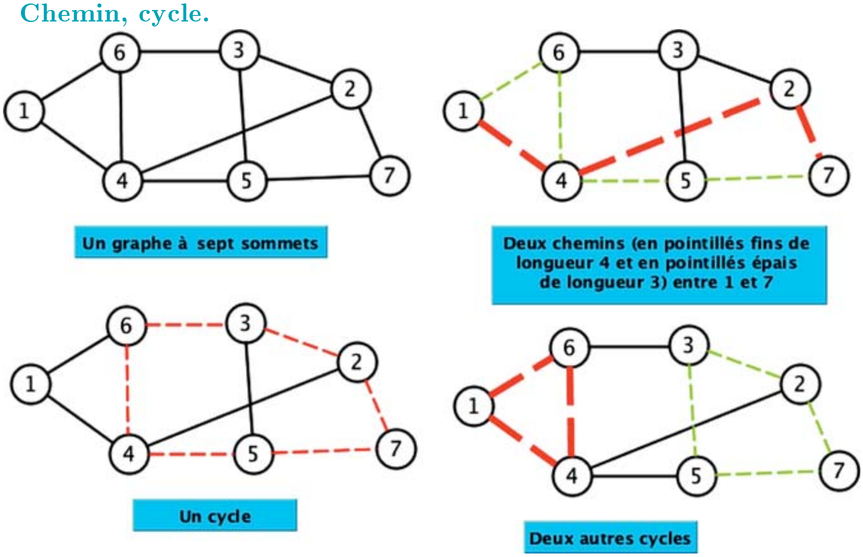
Figure 1: Un graphe avec cinq *sommets* et six *arêtes*

$\langle u, v \rangle$ est l'*arête* entre u et v.

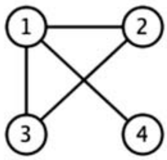
Voisins d'un sommet, degré d'un sommet.

Deux sommets u et v sont *voisins* si le graphe contient l'arête $\langle u, v \rangle$. Par exemple, dans le graphe de la figure 1, les sommets 1 et 5 sont voisins, comme les sommets 5 et 3, ainsi que 3 et 4, etc. En revanche, 2 et 4 ne sont *pas* voisins (il n'y a pas d'arête entre 2 et 4). Le *degré* d'un sommet dans un graphe est son nombre de voisins. Par exemple, le sommet 5 est de degré 4, le sommet 1 est de degré 1.

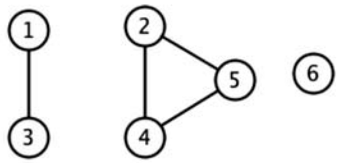
Chemin, cycle.



Graphe connexe. Un graphe est *connexe* s'il existe un *chemin* entre chaque paire de sommets.

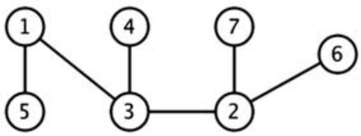


Un graphe connexe

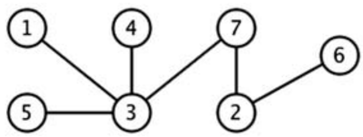


Un graphe à six sommets non connexe

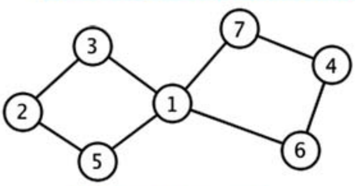
Arbre, arbre couvrant un graphe. Un *arbre* est un graphe connexe sans cycle.



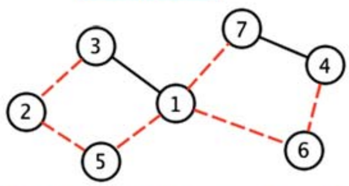
Un arbre (graphe connexe et sans cycle)



Un autre arbre

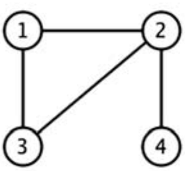


Un graphe (connexe)

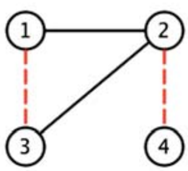


Un arbre (pointillés) couvrant le graphe

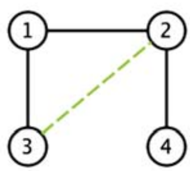
Couplage. Un *couplage* est un ensemble d'arêtes qui n'ont aucun sommet en commun.



Un graphe



Un couplage
(ici de taille maximale)



Un autre couplage
(qui ne peut pas être agrandi)

1 Présentation

De manière directe ou indirecte, chacun d'entre nous utilise de nos jours un logiciel, un produit ou un service numérique. Malgré cela, certaines personnes pensent que l'informatique n'est qu'un outil plus ou moins neutre, une sorte de gadget juste bon à faire la fortune de quelques étudiants qui ont débuté dans un garage en Californie. Si c'est votre cas, je vous propose une expérience de pensée. Imaginez que l'on arrête, d'un coup, tous les ordinateurs du pays. Tous les ordinateurs, cela veut dire bien sûr les ordinateurs de bureau mais aussi ceux qui font vivre les réseaux (Internet, téléphone), les serveurs de données (des banques, des assurances, des réseaux de distribution et de vente, de la santé, etc.), mais aussi les ordinateurs embarqués comme les smartphones, les outils de type GPS, etc. On imagine bien que nos activités et nos vies seraient gravement perturbées, pour le moins.

Il faut bien se rendre compte que tous ces services informatiques, ces logiciels, ces réseaux, ces objets connectés ont été imaginés puis conçus, programmés, testés, déployés, commercialisés, maintenus, améliorés, sécurisés, etc., par des armées d'informaticiens qui œuvrent souvent dans l'ombre.

L'image que le grand public a d'une profession se construit souvent à partir de personnages vus dans des films, généralement des fictions hollywoodiennes ou des séries télévisées. Quel est l'informaticien type dans ces représentations ? Généralement un homme, plutôt jeune, plutôt socialement inadapté, se nourrissant de pizzas, buvant des litres de boissons gazeuses sucrées dans son antre dans lequel règne un chaos de machines, de fils et de cartes électroniques. De ces représentations fictives, beaucoup de personnes retiennent que son travail se réduit à la programmation et qu'il passe ses journées à taper frénétiquement sur un clavier. C'est loin d'être le cas pour nombre d'entre eux (même si c'est vrai pour certains). La programmation est bien sûr une activité informatique mais ce n'est pas la seule et ce n'est pas la première (en tout cas, ça ne devrait pas l'être). Avant d'écrire le code d'un logiciel, il faut savoir comment faire les opérations qui sont demandées. Il faut un plan.

L'analogie (classique) avec la cuisine. En cuisine, si vous voulez faire un plat il vous faut une recette (le « plan » du plat). Vous suivez alors les instructions ligne à ligne pour préparer ce dont vous avez envie à partir des ingrédients. Ce livre va vous révéler les secrets de quelques recettes informatiques. C'est ce que l'on appelle un *algorithme*, c'est-à-dire une méthode (la recette) qui permet de construire une solution (cuisiner un plat) à partir de données (les ingrédients) en suivant les instructions pas à pas. Cet algorithme pourra ensuite être programmé puis intégré dans un ensemble plus vaste qui va constituer une solution informatique globale (le repas complet). La comparaison entre les algorithmes et la cuisine est un véritable cliché, mais elle est tellement parlante qu'il serait dommage de s'en passer.

Les livres sur les algorithmes sont nombreux mais ils s'adressent aux experts (étudiants en informatique, ingénieurs, etc.). Dans ces ouvrages, ils sont décrits dans des langages proches de ceux de programmation, ce qui est normal puisqu'ils s'adressent à des gens qui vont devoir les programmer. Ils rebutent donc le grand public non initié, comme tout livre technique « pointu » dans quelque domaine que ce soit. Les algorithmes sont nombreux, bien trop nombreux et diversifiés pour une courte présentation. Il y a des algorithmes pour traiter des textes (par exemple pour rechercher de manière efficace un mot dans un long fichier), pour traiter des images (construire des images de synthèse ou analyser de manière automatique des photos), pour traiter des données : les compresser (en réduire le « volume »), les sécuriser (cryptographie), les analyser (extraction de données « pertinentes » dans de grandes masses de données), etc.

Vulgarisation algorithmique. L'ambition de ce livre est de vous présenter bon nombre de résultats et d'algorithmes en vous décrivant les idées principales de ces méthodes, sans utiliser de jargon technique. Il suffira de vous laisser guider par les explications pour en comprendre la « substantifique moelle ». De nombreux exemples et illustrations éclaireront le propos. Inutile d'avoir un ordinateur pour lire ce livre mais quelques feuilles de papier, un crayon et une gomme vous seront utiles si vous voulez reproduire les exemples ou en créer d'autres (ce que je vous conseille de faire si vous désirez approfondir le sujet). Je sais d'expérience qu'un tel livre ne se lit pas forcément comme un roman, de manière linéaire, de la première à la dernière page, dans l'ordre. Peut-être lirez-vous quelques passages, puis vous poserez l'ouvrage, pour le reprendre plus tard. Pour faciliter ce cheminement, les chapitres sont volontairement courts et en grande partie indépendants, de manière à ce que vous puissiez les lire à votre rythme. De plus, les passages qui demandent un peu plus de calculs ou nécessitent des raisonnements un peu plus complexes sont dans des parties bien identifiées, des zones colorées, comme ci-dessous.

Zone orange

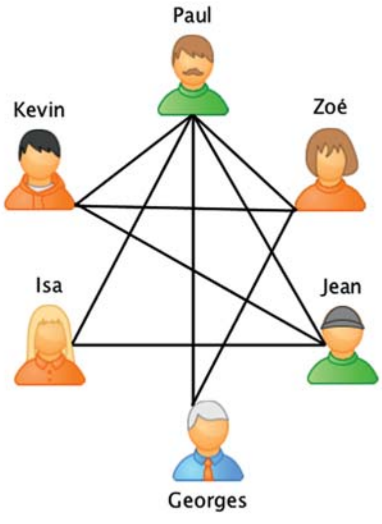
Une zone orange contient des raisonnements moins immédiatement accessibles.

Zone rouge

Une zone rouge contient des explications ou des calculs encore plus pointus.

La difficulté de lecture d'une zone orange ou rouge dépend forcément de vos connaissances préalables sur le sujet ou de votre aisance naturelle à appréhender de nouveaux concepts. Un conseil : essayez de les lire. Si vous éprouvez trop de difficultés, sautez-les et revenez-y plus tard.

Les graphes. Le choix fait ici est de se focaliser sur la manipulation d'un objet abstrait qui est le *graphe*. La raison est simple : les graphes sont partout. Imaginez une soirée organisée par un ami, appelons-le Paul. Parmi les invités, il y a des gens que vous connaissez bien, vos amis communs avec Paul et les autres. Certains d'entre eux se connaissent, même si vous, vous ne les connaissez pas. Faisons une expérience. Si deux personnes sont amies, elles vont se relier l'une à l'autre par un long fil de laine. Par exemple, si Zoé a trois amis présents à cette soirée, elle sera à l'extrémité de trois fils la reliant à chacun de ses trois amis. La longueur des fils importe peu. Essayez de visualiser la situation. Pour vous aider, voici un schéma d'une telle (petite) rencontre. Un fil de laine est représenté par un trait.

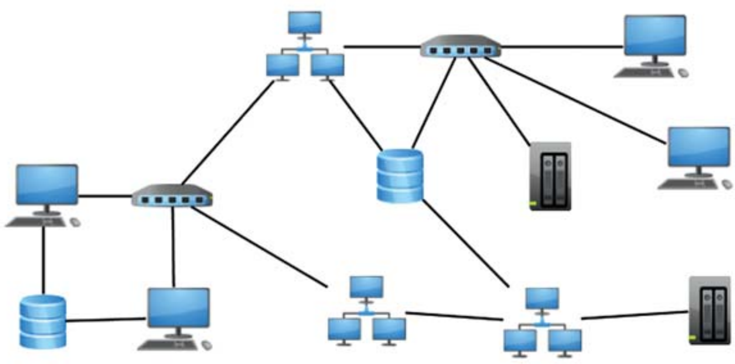


Le nombre de fils peut être très important. Si la soirée réunit un groupe de quinze amis qui se connaissent très bien, chaque personne sera ainsi à une des deux extrémités de quatorze fils. Même si l'appartement est grand, ils vont avoir du mal à se déplacer car le nombre total de fil sera de… (je vous laisse y réfléchir, la réponse est dans un des chapitres).

Formellement, un graphe est la donnée d'un ensemble d'éléments, nommés les *sommets*–dans notre exemple ce sont les personnes présentes à la soirée–et d'un ensemble de *relations* entre ces éléments, nommées les *arêtes*. Dans notre exemple, une arête est matérialisée par un fil de laine et représente une relation d'amitié entre les deux personnes reliées.

Ces dernières décennies, les graphes ont été utilisés dans l'industrie et la science pour représenter, modéliser, manipuler toutes sortes d'objets, de natures différentes. Voici quelques exemples.

– Des pages web : chaque page est un sommet et une relation (orientée) est placée d'une page A vers une page B si la page A contient un lien qui permet d'aller directement sur la page B. Les moteurs de recherche utilisent ce type de modélisation pour présenter les pages les plus pertinentes, répondant au mieux aux critères de recherche des internautes.
– Des réseaux informatiques : chaque ordinateur est un sommet et chaque liaison directe entre deux machines est représentée par une arête. Cette « carte » (qui peut être changeante) sert de base aux opérations de routage (trouver une route entre deux points pour acheminer un message).



– Des réseaux sociaux, réels ou virtuels, issus du domaine professionnel, associatif, politique, économique, etc. La représentation sous forme de graphe de ces réseaux permet par exemple de les analyser en déterminant des communautés ou les personnes qui sont les plus influentes.
– Des réseaux de distribution de marchandises ou d'énergie. Ici la représentation des points de production, de transformation, de distribution, etc. (sommets), et les moyens d'acheminement des marchandises (arêtes) sert à organiser et à optimiser les flux de production et de distribution.
– Des plans. Lorsque vous prenez l'avion, la brochure publicitaire de la compagnie que vous trouvez dans la pochette du siège représente ses liaisons aériennes sous forme de graphe : un aéroport est un sommet et une arête est une liaison. Un plan de métro est aussi un graphe.
– Des relations entre des informations. Les programmes informatiques manipulent des données, parfois très volumineuses. Il faut les organiser de manière à ce que les ajouts, suppressions ou modifications soient très rapides. Cela est le domaine des *structures de données* où les éléments sont mis dans des cellules, reliées par des relations pour les organiser. Le tout peut être vu comme un graphe.

Les graphes sont virtuellement partout. Il en existe d'ailleurs de plusieurs types (orientés, pondérés, hypergraphes, multigraphes, etc.) pour tenir compte des spécificités des situations. Ils sont exploités pour organiser des structures (optimiser les réseaux de distribution), pour essayer de les comprendre (dynamique des réseaux sociaux), pour aider à les sécuriser (chercher des routes alternatives lors de pannes ou d'attaques), pour rechercher des informations (dans des structures de données), etc.

Pour certaines personnes (des ingénieurs par exemple), les graphes sont un outil, pour d'autres c'est un objet d'étude et de recherche (plusieurs thèses sur les graphes sont soutenues chaque année en France). Cet outil abstrait est très simple à décrire et consiste comme nous l'avons dit en un ensemble d'éléments et de relations entre ces éléments. Les algorithmes qui sont au menu (pour rester dans l'analogie culinaire) de ce livre manipulent des graphes. C'est ce que nous découvrirons pas à pas.

Les apparences sont trompeuses. La recherche en marche. Nombre de problèmes de graphes, simples à décrire, sont, en pratique, très difficiles à résoudre. Ce point est passionnant et, pour tout dire, finalement assez intriguant. Pour contourner cette difficulté, des programmes de recherche universitaires ont été mis en œuvre. Ce livre présente dans quelques chapitres une voie explorée, nommément les *algorithmes d'approximation*. Certains sont suffisamment simples pour pouvoir être décrits ici. Ces méthodes s'appuient souvent sur celles, plus classiques, vues dans les premiers chapitres.

Ordre de lecture des chapitres. Il est plutôt conseillé de lire les chapitres dans l'ordre des numéros mais ce n'est pas une obligation. Comme cela a été mentionné plus haut, ils ont été écrits pour être en grande partie indépendants. Cependant il est préférable de lire le chapitre 2 en premier car il présente la notion centrale de graphe, donne des illustrations pour vous familiariser avec le sujet et présente quelques résultats qui seront utiles dans d'autres chapitres. Les notions importantes pour comprendre un chapitre sont rappelées, même si elles ont été vues dans un précédent chapitre, de manière à vous éviter de faire des retours. Cela induit quelques redites mais vous permettra aussi de les revoir sous un autre angle, avec de nouveaux exemples. Pour lire les derniers chapitres, il est fortement conseillé d'avoir lu auparavant le chapitre 14.

Bon voyage. Ce livre est pour vous, même (et surtout) si vous ne connaissez rien à ces domaines. Bon voyage dans le monde des graphes et des algorithmes de graphes.

2 Un graphe. Qu'est-ce que c'est ?

Dans ce chapitre, nous allons voir ensemble des éléments qui seront utiles tout au long de ce livre. Nous verrons qu'il est facile de dessiner un graphe et de le manipuler « à la main ». Mais commençons par la base. Un *graphe* est composé de deux choses :

- un ensemble d'éléments, qui sont appelés les *sommets du graphe* ;
- un ensemble de *relations* entre ces éléments. Plusieurs types de relations sont possibles mais nous allons nous focaliser principalement (pas exclusivement) dans ce livre sur les relations non orientées qui sont nommées les *arêtes du graphe*. Rien qu'avec cela, nous avons déjà largement de quoi faire une belle promenade.

Voyons tout cela un peu plus en détail avant de donner quelques exemples.

Qu'est-ce qu'un sommet ? Encore une fois, un sommet est simplement un élément, faisant partie d'un ensemble : l'ensemble des sommets d'un graphe. Dans ce livre, nous allons nous restreindre aux graphes ayant un nombre fini de sommets (les graphes infinis sont aussi étudiés mais nous n'en parlerons pas du tout). Chaque sommet va avoir un nom, un identifiant, qui lui est propre et qui permet de le distinguer des autres. Ce sera souvent un numéro dans les divers exemples qui seront donnés. Mais ce n'est pas obligatoire. Si vous voulez représenter le graphe des stations de métro de la ville de Paris, les sommets peuvent avoir les noms des stations.

Qu'est-ce qu'une arête ? Une arête entre les sommets u et v d'un graphe sera notée $\langle u, v\rangle$ dans ce livre. L'arête $\langle u, v\rangle$ traduit le fait que les deux éléments/sommets u et v sont en relation l'un avec l'autre, ils sont *liés* par cette arête. L'ordre de la relation n'est pas important ici. L'arête $\langle u, v\rangle$ est la même que $\langle v, u\rangle$. Dans certains ouvrages, une arête $\langle u, v\rangle$ est notée uv ou $\{u, v\}$ ou $R(u, v)$. La nature ou l'origine d'une relation entre u et v importe peu ici. Le point important pour la définition d'un graphe est de savoir s'il y a ou non une arête entre u et v.

Entre chaque paire de sommets, il y a soit une soit zéro arête, ni plus, ni moins. Les graphes que nous allons manipuler ont donc un ensemble fini de sommets, par exemple : $\{1, 2, 3, 4, 5, 6\}$. Entre chaque paire de sommets, il y a soit une arête soit zéro arête. Une arête sera toujours pour nous dans ce livre une relation entre exactement *deux* sommets (il n'y a pas d'arête entre u et lui-même).

Un graphe sera noté $G = (V, E)$ où V représente l'ensemble des sommets (*vertices* en anglais) et E représente l'ensemble des arêtes (*edges* en anglais).

Représentation graphique des graphes. Un énorme avantage de la théorie des graphes par rapport à d'autres domaines des mathématiques est qu'ici, la représentation graphique des objets est simple (s'ils ne sont pas trop gros) et sert très souvent de support à l'intuition. Nous allons pleinement en profiter. Les sommets du graphe vont être représentés par des points sur la feuille de papier. Un sommet va volontairement être grossi de manière à pouvoir voir son nom, son numéro. Une arête entre deux sommets u et v sera représentée par un trait entre les deux sommets. Les traits ne sont pas nécessairement droits et peuvent se croiser (sauf dans le chapitre 7 de ce livre). Il faut impérativement représenter toutes les arêtes, sans ambiguïté. Le placement des sommets n'est pas imposé. L'essentiel est de bien distinguer tous les éléments du graphe.

ILLUSTRATION

Considérons le graphe $G = (V, E)$ représenté sur la figure 2.1.

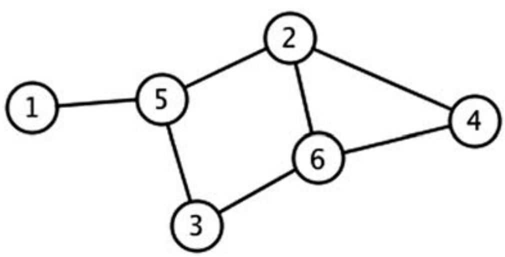


Figure 2.1: Votre premier graphe

Ses sommets sont 1, 2, 3, 4, 5 et 6 et il a sept arêtes. Il y a par exemple les arêtes $\langle 2, 4\rangle$, $\langle 3, 6\rangle$. En revanche, il n'y a pas d'arête entre 2 et 3, ni entre 1 et 4. La représentation graphique est arbitraire. Ce graphe peut aussi être dessiné comme sur la figure 2.2. Ce sont exactement les mêmes sommets et les mêmes arêtes mais représentées sous la forme d'un autre dessin. Notez au passage que sur la figure 2.2, les arêtes $\langle 2, 4\rangle$ et $\langle 3, 5\rangle$ se croisent, ce qui n'a pas d'importance.

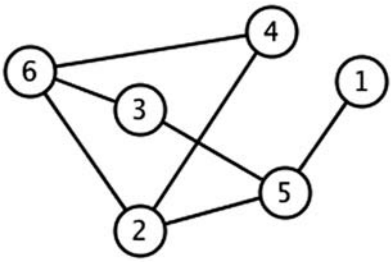
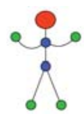


Figure 2.2: Une autre représentation graphique du graphe de la figure 2.1

Des voisins. Le degré d'un sommet. Si un graphe G contient l'arête $\langle u, v\rangle$, alors u et v sont dits *voisins* (l'un de l'autre) ; ils partagent la même arête, ils sont liés directement l'un à l'autre. Par exemple dans le graphe de la figure 2.2, le sommet 6 est voisin de 4 (mais aussi de 3 et de 2), le sommet 1 a comme unique voisin le sommet 5. Le nombre de voisins d'un sommet u dans un graphe G est son *degré* noté $deg_G(u)$. Par exemple, le sommet 3 est de degré 2, le sommet 1 est de degré 1.



Les chemins d'un graphe. On constate que tous les sommets ne sont pas forcément voisins les uns des autres. Par exemple, dans le graphe de la figure 2.2, les sommets 2 et 3 ne sont pas voisins. Mais on peut, malgré tout, « aller » de 2 vers 3 (ou de 3 vers 2) en suivant plusieurs arêtes, par exemple la suite d'arêtes : $\langle 2, 5\rangle$, $\langle 5, 3\rangle$ ou la suite, plus longue, $\langle 2, 4\rangle$, $\langle 4, 6\rangle$, $\langle 6, 3\rangle$. Cela nous conduit vers la notion de *chemin* qui est très importante. On ne va pas en donner une définition formelle mais une idée intuitive. Pour cela, imaginons un personnage que l'on nommera Graphix (on le retrouvera plusieurs fois dans cet ouvrage) qui a la capacité de se promener sur un graphe. Un graphe $G = (V, E)$ contient un chemin entre u et v si Graphix est capable, en partant du sommet u et en passant de voisin en voisin (en suivant des arêtes), d'aller jusqu'au sommet v. Lors de ce parcours, Graphix ne doit pas passer plusieurs fois par la même arête ou le même sommet (c'est un chemin qui est dit *élémentaire*). La *longueur* d'un chemin est son nombre d'arêtes, c'est-à-dire le nombre de pas que fait notre personnage pour aller d'une extrémité à l'autre.

Cas particulier : chemin de longueur 0. Par convention, il y a toujours un chemin entre n'importe quel sommet u et lui-même, sans arête, de longueur 0.

ILLUSTRATION

La figure 2.3 (a) représente un graphe. La figure 2.3 (b) représente un chemin entre 1 et 9, de longueur 6 (car il contient les 6 arêtes en pointillés). Ce chemin peut être identifié par la suite de ses sommets, ici $1, 2, 6, 10, 7, 5, 9$ ou, comme cela est représenté graphiquement sur la figure, par la suite de ses arêtes, ici $\langle 1, 2\rangle$, $\langle 2, 6\rangle$, $\langle 6, 10\rangle$, $\langle 10, 7\rangle$, $\langle 7, 5\rangle$, $\langle 5, 9\rangle$.

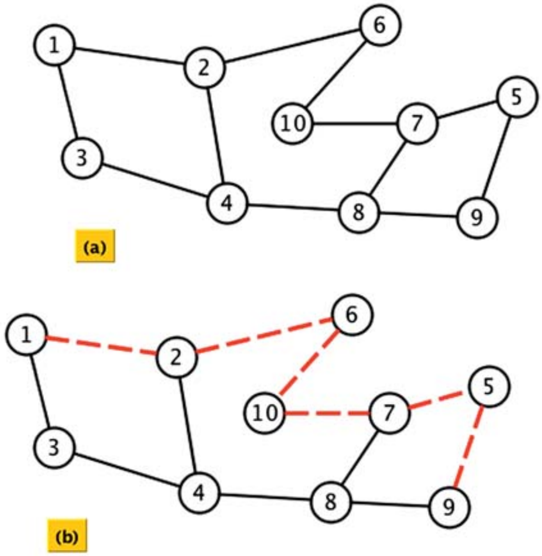


Figure 2.3: (a) Un graphe. (b) Un chemin de longueur 6 entre les sommets 1 et 9

Sur la figure 2.4, un autre chemin entre 1 et 9 est représenté en pointillés ; il est de longueur 4. Il n'y en a pas de plus court entre 1 et 9. En revanche, le chemin $1, 3, 4, 8, 9$ est lui aussi de longueur 4. Voici quelques autres chemins de ce graphe : $2, 4, 8, 7$ (de longueur 3), $3, 1, 2, 4$ (de longueur 3 aussi), $8, 7$ (de longueur 1).

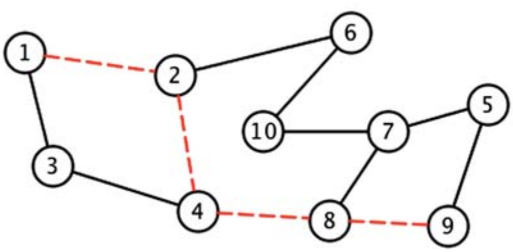


Figure 2.4: Un chemin de longueur 4 entre les sommets 1 et 9

Graphe connexe. Pour aborder la notion de connexité, examinons le graphe de la figure 2.5. Il est composé de sept sommets et de six arêtes. C'est bel et bien un graphe. Il y a bien au plus une arête entre chaque paire de sommets. Mais ce graphe n'est pas « en un seul morceau ». Si Graphix part d'un sommet quelconque, il ne pourra pas rejoindre n'importe quel autre sommet en suivant un chemin. S'il part, par exemple, du sommet 4, il n'y a aucun chemin pour aller jusqu'en 1 (ou en 3 ou en 2). Dans ce cas, on dit que le graphe n'est pas connexe.

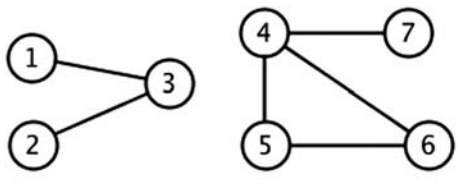


Figure 2.5: Un graphe non connexe à six sommets (avec deux composantes connexes)

Un graphe G est *connexe* si pour toute paire de sommets u et v, il existe un chemin dans G entre u et v. Dit autrement, G est connexe si GRAPHIX peut aller de n'importe quel sommet à n'importe quel autre sommet en suivant un chemin de G. Par exemple, le graphe de la figure 2.3 (a) est connexe. On imagine bien que si le graphe représente un réseau, la connexité est une notion centrale.

Les cycles. Un *cycle* dans un graphe G est un chemin dont les deux extrémités sont les mêmes. Si GRAPHIX parcourt un cycle en partant d'un sommet u, il sera de retour en u à la fin de son périple circulaire. On suppose ici aussi, comme pour les chemins, que GRAPHIX ne passe pas plusieurs fois par une arête ou par un sommet, sauf son point de départ où il revient à la fin. La *longueur* d'un cycle est son nombre d'arêtes. La figure 2.6 présente en pointillés un cycle de longueur 8 de G que l'on note $2, 6, 10, 7, 5, 9, 8, 4, 2$ (le 2 en début *et* en fin de séquence indique que l'on a affaire à un cycle). Cette séquence aurait aussi bien pu s'écrire $10, 7, 5, 9, 8, 4, 2, 6, 10$ car le point de départ n'a pas d'importance dans quelque chose de circulaire. Voici d'autres cycles de ce même graphe : $7, 5, 9, 8, 7$ (de longueur 4), $2, 6, 10, 7, 8, 4, 2$ (de longueur 6) ou même $1, 2, 6, 10, 7, 5, 9, 8, 4, 3, 1$ qui contient les dix sommets du graphe et qui est donc de longueur 10.

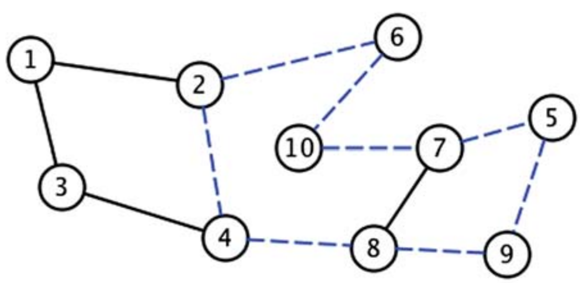


Figure 2.6: Un cycle de longueur 8

Attention, la séquence suivante n'est pas considérée comme un cycle : $1, 2, 1$ car si GRAPHIX la suit, il repasse deux fois par la même arête $\langle 1, 2 \rangle$.

Les arbres. Les arbres constituent une famille importante de graphes que l'on va retrouver dans plusieurs chapitres. Un arbre est un graphe connexe et sans cycle.

C'est-à-dire qu'il est composé d'un seul « morceau » et qu'il ne contient pas de cycle. Voici quelques propriétés des arbres.

– Tout arbre à n sommets a nécessairement exactement $n-1$ arêtes.
– Un arbre T est un graphe connexe (par définition) mais pour n'importe quelle paire de sommets u et v, il existe exactement un chemin entre u et v dans T.
– En supprimant n'importe quelle arête de T, le graphe obtenu n'est plus connexe (comme dans la nature, les arbres de la théorie des graphes sont « fragiles »).

ILLUSTRATION

Examinons ces propriétés sur l'exemple de la figure 2.7. Ce qui est représenté est bien un arbre : c'est bien un graphe connexe et il ne possède pas de cycle. On compte $n = 10$ sommets et $m = 9$ arêtes. On a donc bien $m = n - 1$. De plus, n'importe quelle paire de sommets n'est reliée *que* par un seul chemin. Par exemple, si GRAPHIX veut aller de 1 jusqu'à 4, il est obligé de suivre le chemin $1, 3, 2, 4$. Il n'y en a pas d'autre. Vous pouvez aisément vérifier cette propriété pour n'importe quelle paire d'extrémités.

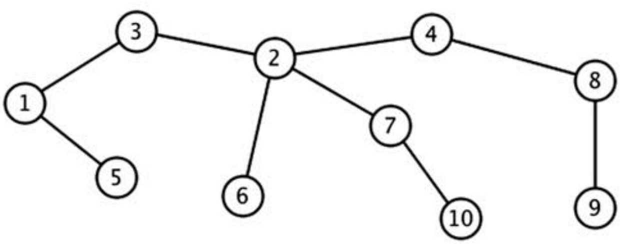


Figure 2.7: Un arbre

La dernière propriété énoncée indique que si une arête de T est supprimée, alors le résultat n'est plus connexe. Par exemple, en supprimant l'arête $\langle 2, 4\rangle$, le graphe obtenu est composé de deux « morceaux » (composantes) connexes : d'un côté les sommets $4, 8, 9$ et de l'autre les sept autres sommets. Autre exemple : en supprimant $\langle 7, 10\rangle$, le sommet 10 est isolé de tous les autres. Chaque arête possède cette propriété de couper le graphe en deux parties (pas forcément de même taille) si elle est supprimée. Je vous invite à vérifier par vous-même ces propriétés sur les trois arbres de la figure 2.8.

Additionnons les degrés de tous les sommets d'un graphe. Considérons un graphe $G = (V, E)$ quelconque, connexe ou non. Faisons la somme des degrés (nombre de voisins) de tous ses sommets. Quel est le résultat ? Je vous invite à y réfléchir avant de lire la suite.

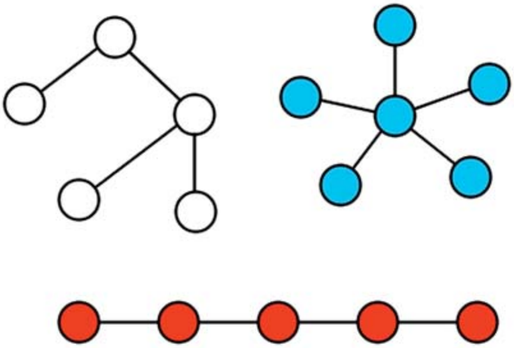


Figure 2.8: Trois arbres

Cette somme vaut toujours $2m$ où m est le nombre d'arêtes de G. Pourquoi ? Car en faisant cette somme, chaque arête $\langle u, v\rangle$ est comptée *exactement deux fois* : en additionnant le degré de u, elle est comptée une fois et en additionnant le degré de v, elle est comptée une autre fois. Ensuite, elle ne sera plus prise en compte (car elle n'a que deux extrémités).

ILLUSTRATION

Illustrons cela sur le graphe de la figure 2.9. Il a $m = 11$ arêtes en tout. Comptez le nombre de voisins de chaque sommet u et vous obtiendrez son degré. Par exemple, le degré du sommet 1 est 3 (car il a trois voisins) ; le degré de 3 est 5, etc. Faisons-en la somme : $3 + 2 + 5 + 3 + 3 + 3 + 3 = 22$, c'est-à-dire précisément deux fois le nombre d'arêtes. Par exemple, l'arête $\langle 3, 7\rangle$ contribue pour une unité dans le degré du sommet 3, pour une unité dans le degré du sommet 7 et nulle part ailleurs. Il en va de même pour chacune des autres arêtes, ce qui explique le résultat final.

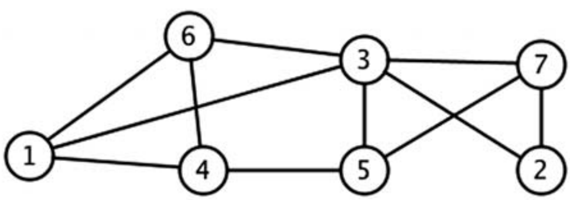


Figure 2.9: Un graphe

Les graphes extrêmes. Quel est le nombre *minimal* d'arêtes que peut contenir un graphe ayant n sommets ? Réfléchissez quelques secondes à la réponse. Beaucoup de gens ont tendance à répondre spontanément qu'un graphe a au minimum $n - 1$ arêtes (surtout après avoir lu le passage sur les arbres). Il n'en est rien. Un graphe peut

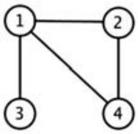
Quelques revues.

- Le magazine *Tangente* (pôle éditions) propose parfois des articles sur les graphes ou la combinatoire. En particulier, le hors-série numéro 54 de ce magazine y est entièrement consacré.
- La rubrique de Jean-Paul Delahaye dans le magazine *Pour la Science* traite parfois de problèmes de graphes. Ces rubriques sont éditées sous forme de livres aux éditions Belin (voir plus haut).
- À côté de ces quelques revues destinées au grand public, auxquelles on peut ajouter *La Recherche*, il existe de nombreuses revues professionnelles, universitaires, consacrées aux mathématiques discrètes, à la combinatoire et aux graphes. Les prix et le niveau de ces revues font qu'elles sont réservées aux laboratoires de recherche. Vous pourrez peut-être les consulter dans la section recherche d'une bibliothèque universitaire. Inutile de les chercher dans le kiosque à journaux du coin de la rue.

Comment représenter un graphe dans la mémoire d'un ordinateur ? Après avoir manipulé des graphes « à la main » dans les chapitres précédents, certains lecteurs peuvent se demander comment un tel objet peut être représenté dans la mémoire d'un ordinateur. Il existe plusieurs moyens pour cela. Pour les décrire, il faudrait entrer dans des détails pointus de programmation qui ne sont pas l'objet de ce livre. Nous allons simplement en détailler un assez simple à comprendre.

Prenons un graphe $G = (V, E)$ à n sommets. Chaque sommet doit être associé à un numéro unique entre 1 et n (ou entre 0 et $n - 1$). Il faut ensuite créer une matrice M, $n \times n$ sous la forme d'un *tableau* dans la mémoire de la machine. Les arêtes sont codées par des 1 dans cette matrice avec la règle suivante : si G contient l'arête $\langle i, j\rangle$, alors $M[i, j] = M[j, i] = 1$, sinon $M[i, j] = M[j, i] = 0$. Avec cette règle, la matrice est *symétrique*, c'est-à-dire que $M[i, j] = M[j, i]$. On note aussi que la diagonale n'est composée que de 0 ($M[i, i] = 0$ pour tout i).

Illustrons cela avec le graphe suivant dont tous les sommets ont déjà un numéro entre 1 et 4.



La matrice M qui correspond à ce graphe est :

	1	2	3	4
1	0	1	1	1
2	1	0	0	1
3	1	0	0	0
4	1	1	0	0

Examinons quelques cases. $M[2, 3] = M[3, 2] = 0$ car le graphe ne contient pas l'arête entre les sommets 2 et 3. En revanche, $M[3, 1] = M[1, 3] = 1$ car il contient une arête entre 1 et 3.

Pour savoir si les sommets i et j sont voisins, il suffit de lire le contenu de la case $M[i, j]$ (ou $M[j, i]$) et de la comparer à la valeur 1, ce qui est très rapide.

Codages d'autres graphes avec une matrice. L'avantage de ce codage est qu'il permet de représenter un graphe *orienté* avec un principe similaire : si G a un *arc* (i, j), alors $M[i, j] = 1$. En revanche, si G n'a pas l'arc (j, i), alors $M[j, i] = 0$. La matrice d'un graphe orienté n'est donc pas forcément symétrique.

Il est aussi facile et pratique de représenter un graphe *pondéré* (orienté ou non) : la case $M[i, j]$ reçoit la valeur du poids de l'arc ou de l'arête entre i et j (et la valeur 0 s'il n'y a pas de lien entre les deux).

Représentation d'autres éléments en mémoire. Lors du parcours en profondeur ou en largeur, les sommets sont coloriés au fur et à mesure qu'ils sont visités (pour éviter de les visiter plusieurs fois). Dans le problème de la coloration vu au chapitre 15, l'objectif final est d'attribuer une couleur aux sommets. Colorier les sommets est une activité courante et très utile. Mais comment représenter ces couleurs ? Chaque couleur est représentée par un entier : $1, 2, \ldots$ Il suffit de prendre un tableau C à une seule ligne et n colonnes. La valeur de la case $C[i]$ représente alors la couleur du sommet i. Il est facile de modifier la couleur d'un sommet en allant modifier le contenu de la case qui lui est associée.

Plus généralement, il est important de pouvoir stocker des données complexes de manière à ce qu'elles soient facilement accessibles, aussi bien en lecture (lire une valeur) qu'en écriture (ajouter ou modifier une valeur). Pour cela, il faut faire appel à des *structures de données* plus compliquées à manipuler que de simples matrices, dont certaines sont organisées sous forme d'arbres. On en revient encore une fois aux graphes, mais cette fois bien cachés dans les entrailles des logiciels.