Introduction à la théorie des graphes

La hiérarchie de Toussaint

Godfried Toussaint (1944-2019) était un informaticien canadien. Il est considéré comme le père de la géométrie algorithmique, c'est-à-dire l'étude et la production de processus systématiques permettant la résolution de problèmes géométriques. Ses travaux de recherche concernent de nombreux domaines (théorie des graphes, reconnaissance de forme, planification de mouvement, traitement d'images, théorie des nœuds, liaison mécanique, problème de la galerie d'art, triangulation, problème du plus grand cercle vide, construction à la règle et au compas, apprentissage automatique, informatique musicale, etc.). Il a introduit le graphe de voisinage relatif (*Relative Neighborhood Graph*) en 1980 et a démontré qu'il contient l'arbre couvrant minimal (*Minimal Spanning Tree*) et est continu dans la triangulation de Delaunay (*Delaunay Triangulation*). D'autres graphes remarquables ont plus tard été ajoutés à cette succession d'inclusions qu'on appelle désormais la hiérarchie de Toussaint.

1 Généralités sur les graphes

Le but de ce chapitre est d'introduire le vocabulaire de base de la théorie des graphes qui sera utilisé dans les chapitres suivants.

1.1 Premières définitions

Les graphes sont des objets mathématiques abstraits qui modélisent les dessins formés de réseaux de lignes reliant des points. Plus précisément, on utilise la définition suivante.

Définition 1

*Un **graphe** est un couple $G = (V, E)$ constitué d'un ensemble V non vide et fini, et d'un ensemble E de paires d'éléments de V. Les éléments de V sont appelés les **sommets** du graphe G, ceux de E sont appelés les **arêtes** du graphe G. Si $e = \{v, v'\}$ est une arête de G, on dit que les sommets v et v', qui sont les **extrémités** de l'arête e, sont **voisins** dans le graphe G.*

En anglais, les sommets sont appelés «*Vertices*» et les arêtes sont appelées «*Edges*» (et un graphe est appelé «*Graph*»), d'où les notations utilisées.

On peut dénombrer le nombre de graphes possibles ayant un ensemble fixé de sommets.

Propriété 1

On note $n = \mathrm{card}(V) \in \mathbb{N}^*$ un nombre de sommets. Puisqu'une arête est une 2-combinaison de V, il y a $\binom{n}{2}$ arêtes possibles ayant ses extrémités dans V. Puisque E est une partie de l'ensemble des 2-combinaisons de V, il y a $2^{\binom{n}{2}}$ graphes possibles ayant V pour ensemble de sommets.

Par exemple, si on fixe seulement 10 sommets, il existe environ $3,5 \times 10^{13}$ graphes possibles ayant cet ensemble de sommets. Ce qui illustre la richesse de la théorie des graphes.

Chaque sommet peut être l'extrémité de plusieurs arêtes, d'une seule arête, ou même d'aucune.

Définition 2

*Un sommet d'un graphe est dit **isolé** lorsqu'il n'a pas de voisin. Un **graphe sans sommet isolé** est donc un graphe tel que chaque sommet est l'extrémité d'au moins une arête.*

1.2 Chemins et connexité

Certains sommets d'un graphe peuvent être reliés par une suite d'arêtes. On définit précisément cette relation par la notion de chemin.

Définition 3

Un ***chemin*** *d'un graphe est une suite non vide et finie de sommets telle que chaque couple de sommets consécutifs soient voisins. Le* ***départ*** *du chemin est son premier sommet, et l'****arrivée*** *du chemin est son dernier sommet. Un chemin est dit* ***élémentaire*** *si aucun sommet ne figure plus d'une fois dans la suite à l'exception du départ et de l'arrivée du chemin qui peuvent coïncider.*

Ainsi, $(v_0, v_1, v_2, \ldots, v_m)$ est un chemin si et seulement si $\{v_i, v_{i+1}\} \in E$ pour tout $i \in [\![0, m-1]\!]$. Son départ est v_0 et son arrivée est v_m. Et ce chemin est élémentaire si et seulement si $v_i \neq v_j$ pour tout couple $(i,j) \in [\![0, m-1]\!]^2$ tel que $i \neq j$.

Un chemin qui n'est pas élémentaire «contient» toujours un chemin élémentaire ayant même départ et même arrivée (qu'on obtient en supprimant les sommets compris entre deux sommets identiques). Ainsi, s'il existe un chemin reliant deux sommets, on peut toujours supposer que ce chemin est élémentaire.

Un chemin élémentaire de $m+1$ sommets est caractérisée par les m arêtes distinctes reliant ses sommets consécutifs.

Définition 4

Deux sommets d'un graphe sont dits ***reliés par un chemin*** *s'il existe au moins un chemin (élémentaire) ayant pour départ et pour arrivée ces deux sommets.*

Savoir lequel des deux sommets est le départ (et donc l'autre est l'arrivée) du chemin n'a pas d'importance. En effet, s'il existe un chemin ayant pour départ un sommet v et pour arrivée un sommet v', alors le chemin obtenu en inversant l'ordre des sommets (avec les mêmes arêtes «parcourues» en sens inverse) a pour départ v' et pour arrivée v. Autrement dit, la propriété d'être reliés par un chemin est symétrique.

De même, on peut démontrer le résultat suivant.

Propriété 2

La propriété d'être reliés par un chemin est transitive : si v et v' sont reliés par un chemin, et si v' et v'' sont reliés par un chemin, alors v et v'' sont reliés par un chemin.

Cependant, deux sommets non voisins ne sont pas toujours reliés par un chemin.

Définition 5

Un graphe est dit ***connexe*** *lorsque deux sommets quelconques sont toujours reliés par (au moins) un chemin.*

Par exemple, un graphe constitué d'un seul chemin est connexe. Évidemment, il existe de nombreux autres exemples de graphes connexes.

Par définition, un sommet isolé n'est relié par un chemin à aucun autre sommet (à l'exception de lui-même). Par contraposée, on obtient le résultat suivant.

Propriété 3

Si un graphe ayant au moins deux sommets est connexe alors il est sans sommet isolé.

1.3 Cycles et triangles

Un sommet (non isolé) peut être relié par un chemin à lui-même à l'aide de n'importe quel chemin dont il est le départ qu'il suffit de parcourir en sens inverse après avoir atteint l'arrivée (par transitivité). Mais il existe des chemins dont le départ est égal à l'arrivée et qui ne «repassent» pas par les mêmes arêtes.

Définition 6

*Un **cycle** d'un graphe est un chemin élémentaire dont le départ et l'arrivée coïncident et passant par au moins deux autres sommets. Un cycle ayant exactement trois sommets est appelé un **triangle**.*

Ainsi, $(v_0, v_1, v_2, \ldots, v_m)$ est un cycle de m sommets si et seulement si les quatre conditions suivantes sont vérifiées :

$$m \geqslant 3, \quad v_0 = v_m, \quad \forall i \in [\![0, m-1]\!],\ \{v_i, v_{i+1}\} \in E \quad \text{et} \quad \forall (i,j) \in [\![0, m-1]\!]^2,\ i \neq j \implies v_i \neq v_j.$$

Un cycle de m sommets est caractérisé par les m arêtes distinctes reliant ses sommets consécutifs. Le choix du départ (et donc de l'arrivée) dans le cycle n'a aucune importance.

Par définition, les triangles sont les plus «petits» cycles (en nombre de sommets).

Définition 7

*Un graphe est dit **triangulé** si chacun de ses cycles d'au moins quatre sommets possède une **corde**, c'est-à-dire une arête ayant pour extrémités deux sommets non consécutifs du cycle.*

Un cycle possédant une corde a toujours au moins quatre sommets (les deux extrémités de la corde et au moins un sommet entre ces extrémités de chaque «coté» du cycle). En particulier, les triangles n'ont pas de cordes.

Un cycle possédant une corde peut être «scindé» en deux cycles plus petits, c'est-à-dire deux cycles ayant chacun un nombre strictement inférieur de sommets. En effet, si l'arête $\{v_0, v_k\}$ est une corde du cycle $(v_0, v_1, v_2, \ldots, v_m)$ alors $k \in \{2, \ldots, m-2\}$ (car $v_0 = v_m$ et v_k ne sont pas consécutifs), $(v_0, v_1, \ldots, v_k, v_0)$ est un cycle de $k+1$ sommets et $(v_k, v_{k+1}, \ldots, v_m, v_k)$ est un cycle de $m-k+1$ sommets.

Dans un graphe triangulé, on peut donc scinder chaque cycle en deux cycles plus petits qui peuvent eux-même être scindés en cycles plus petits, etc., jusqu'à n'obtenir que des triangles.

1.4 Arbres

La notion d'arbre est très importante en théorie des graphes. Elle a été introduite en 1857 par la mathématicien anglais Arthur Cayley (1821-1895) à qui on doit aussi la multiplication des matrices.

Définition 8

*Un **arbre** est un graphe connexe et sans cycle.*

Les arbres sont donc des graphes particuliers qui modélisent les réseaux ramifiés, comme la ramification des branches d'une plante (les arêtes correspondent aux branches et les sommets aux points d'attache ; l'absence de cycle signifie que les branches ne se rejoignent pas en s'éloignant du tronc).

Il existe de nombreuses autres définitions équivalentes des arbres dont quelques unes sont listées ci-dessous.

Théorème 1

Un graphe $G = (V, E)$ est un arbre si et seulement s'il vérifie l'une des conditions équivalentes suivantes :

(i) *G est connexe et sans cycle ;*

(ii) *G est connexe minimal, c'est-à-dire qu'il n'est plus connexe si on lui supprime l'une quelconque de ses arêtes ;*

(iii) *G est sans cycle maximal, c'est-à-dire qu'un cycle est créé si on lui rajoute une arête quelconque ayant pour extrémités deux sommets non voisins ;*

(iv) *G est connexe et $\operatorname{card}(E) = \operatorname{card}(V) - 1$;*

(v) *G est sans cycle et $\operatorname{card}(E) = \operatorname{card}(V) - 1$;*

(vi) *pour toute paire de sommets de G, il existe un seul chemin de G ayant l'un de ces sommets pour départ et l'autre pour arrivée.*

En particulier, un arbre possède toujours une arête de moins que de sommets.

1.5 Sous-graphes

On définit précisément la relation d'inclusion entre graphes par la notion de sous-graphe.

Définition 9

*On dit qu'un graphe $G' = (V', E')$ est un **sous-graphe** d'un graphe $G = (V, E)$ lorsque $V' \subset V$ et $E' \subset E$. Un **sous-graphe couvrant** d'un graphe $G = (V, E)$ est un sous-graphe $G' = (V, E')$, c'est-à-dire un sous-graphe dont les sommets sont tous les sommets de G. Dans ce cas, on dit aussi que le graphe G **contient** le graphe G' ou que le graphe G' est **contenu** dans le graphe G.*

Par définition, si un sous-graphe couvrant est sans sommet isolé alors le graphe qui le contient est aussi sans sommet isolé. De même, la connexité se transmet d'un sous-graphe couvrant au graphe le contenant. Mais les réciproques sont fausses en général.

Définition 10

*Un **arbre couvrant** d'un graphe est un sous-graphe couvrant qui est un arbre.*

Les arbres couvrants sont étudiés pour leurs applications aux réseaux informatiques : ils permettent de définir une liaison directe pour faire circuler de l'information de manière efficace (c'est-à-dire sans qu'elle perde de temps à tourner dans des boucles du réseau).

D'un point de vue mathématique, les arbres couvrants caractérisent les graphes connexes.

Théorème 2

Un graphe est connexe si et seulement s'il contient un arbre couvrant.

On déduit des théorèmes 1 et 2 la minoration du nombre des arêtes suivante.

Corollaire 1

Si un graphe $G = (V, E)$ est connexe alors

$$\mathrm{card}(E) \geqslant \mathrm{card}(V) - 1,$$

avec égalité si et seulement si G est un arbre.

Autrement dit, un graphe connexe ayant n sommets possède toujours au moins $n - 1$ arêtes.

1.6 Graphes planaires

Une manière de représenter géométriquement les graphes abstraits est de les dessiner à l'aide de points et de segments dans le plan euclidien $\mathbb{R}^2 = \{(x, y) \mid x \in \mathbb{R} \text{ et } y \in \mathbb{R}\}$.

Définition 11

*Un **graphe du plan euclidien** est un graphe dont les sommets sont des points de $\mathbb{R}^2$. La **longueur** d'une arête du graphe est la distance euclidienne entre ses extrémités. La **longueur totale** du graphe est la somme des longueurs de toutes ses arêtes.*

Puisque la distance euclidienne $\mathrm{d}(v, v')$ entre deux points de $\mathbb{R}^2$ est égale à la longueur du segment $[v, v']$, on peut représenter géométriquement chaque arête du graphe par le segment reliant ses deux extrémités. Pour rappel, si $v = (x, y)$ et $v' = (x', y')$ alors

$$[v, v'] = \Big\{\big((1-t)x + tx', (1-t)y + ty'\big) \mid t \in [0, 1]\Big\} \quad \text{et} \quad \mathrm{d}(v, v') = \sqrt{(x' - x)^2 + (y' - y)^2}.$$

La notion de longueur permet d'introduire des problèmes d'optimisation. Par exemple, déterminer un cycle de longueur minimale reliant un ensemble fixé de sommets dans $\mathbb{R}^2$ (appelé le problème du voyageur de commerce).

Avec une telle représentation géométrique, deux arêtes distinctes ne sont pas toujours représentés par deux segments disjoints. En particulier, il ne faut pas confondre les points d'intersection des segments représentant les arêtes avec les sommets (qui correspondent seulement aux extrémités des segments).

Définition 12

Un graphe du plan euclidien est dit ***planaire*** *si aucun segment représentant une arête (c'est-à-dire reliant deux sommets voisins) n'en croise un autre. Une* ***face*** *d'un graphe planaire est une région de* $\mathbb{R}^2$ *qui est délimitée par des segments représentant des arêtes et qui ne contient aucun sommet.*

Le nombre de faces est lié aux nombres de sommets et d'arêtes par la formule suivante introduite par le mathématicien suisse Leonhard Euler (1707-1783) en 1758 et démontrée par le mathématicien français Augustin Louis Cauchy (1789-1857) en 1811.

Théorème 3

Si F *désigne l'ensemble des faces d'un graphe* $G = (V, E)$ *planaire et connexe alors :*

$$\mathrm{card}(V) - \mathrm{card}(E) + \mathrm{card}(F) = 1.$$

En particulier, on retrouve que $\mathrm{card}(E) = \mathrm{card}(V) - 1$ pour les arbres planaires (voir le théorème 1).

Cette formule a de nombreuses conséquences. Par exemple, en remarquant que chaque face est délimitée par au moins trois arêtes et que chaque arête délimite au plus deux faces (donc $3\mathrm{card}(F) \leqslant 2\mathrm{card}(E)$), on obtient la majoration du nombre des arêtes suivante.

Corollaire 2

Si un graphe $G = (V, E)$ *ayant au moins trois sommets est planaire et connexe alors*

$$\mathrm{card}(E) \leqslant 3\mathrm{card}(V) - 3.$$

Autrement dit, un graphe planaire connexe ayant $n \geqslant 3$ sommets possède toujours au plus $3n - 3$ arêtes.

Chaque face d'un graphe planaire est délimitée par un cycle ayant un nombre arbitraire de sommets. Les plus petites faces possibles (en nombre de sommets) sont celles délimitées par des triangles.

Définition 13

Une ***triangulation*** *est un graphe planaire dont chaque face est délimitée par un triangle.*

Par définition, un graphe planaire triangulé est une triangulation. Mais la réciproque est fausse en général puisqu'il peut exister dans une triangulation de grands cycles (en nombre de sommets) sans corde qui délimitent des régions subdivisées en plusieurs faces triangulaires (par exemple un cycle de quatre sommets tous reliés à un cinquième sommet «interne»).

2 Modélisation informatique de graphes

Ce chapitre propose une manière de modéliser et de représenter graphiquement des graphes (du plan euclidien) en Python. Des petits exercices sont proposés afin d'assimiler les commandes présentées et les notations adoptées.

2.1 Création de graphiques en Python

Pour placer un point sur un graphique en Python, on peut utiliser les commandes suivantes.

```
import matplotlib.pyplot as plt
plt.plot(4,2,'.r')
plt.show()
```

La première ligne charge le module `pyplot` de la bibliothèque `matplotlib` qui contient des commandes pour tracer des graphiques. La deuxième ligne repère le point de coordonnées $(4, 2)$ à l'aide d'un point rouge. La troisième ligne affiche le graphique construit avec les commandes précédentes dans une nouvelle fenêtre.

On peut changer le style d'affichage des points en modifiant la chaîne de caractères en argument de la commande `plot`. Les différentes formes et couleurs des marqueurs de points sont listées ci-dessous.

forme	caractère
vide	
point	`.`
cercle	`o`
croix +	`+`
croix ×	`x`
étoile	`*`
carré	`s`
diamant	`d`

couleur	caractère
blanc	`w`
bleu	`b`
cyan	`c`
jaune	`y`
magenta	`m`
noir	`k`
rouge	`r`
vert	`g`

Pour placer plusieurs points, on peut utiliser plusieurs commandes `plot`, ou bien une seule avec les listes de chaque coordonnée. Par exemple :

```
plt.plot(1,2,'+b')
plt.plot(2,3,'+b')
plt.plot(4,1,'+b')
plt.show()
```

ou bien

```
plt.plot([1,2,4],[2,3,1],'+b')
plt.show()
```

Pour tracer un segment reliant deux points, il suffit de rajouter le caractère `-` dans le style d'affichage de la commande `plot`. Par exemple :

```
plt.plot([1,2],[3,5],'og-')
plt.show()
```

Les différents styles d'affichage des segments sont listés ci-dessous.

ligne	caractère
continue	`-`
discontinue	`--`
pointillée	`:`

On peut également relier plusieurs points. Par exemple :

```
plt.plot([1,2,3,4,8,8],[4,4,1,5,5,4],'c:')
plt.plot([4,5],[2.5,2.5],'m-')
plt.plot([6,7,7],[3,4,1],'m-')
plt.show()
```

Exercice 1

Tracer le nombre 42 écrit en chiffres romains dans un graphique (le choix du style d'affichage est libre).

2.2 Modélisation des sommets

Chaque point $v = (x, y) \in \mathbb{R}^2$ peut être vu comme la matrice ligne de ses deux coordonnées.

Un ensemble de $n \geqslant 1$ sommets $V = \{v_1, v_2, \dots, v_n\}$ d'un graphe du plan euclidien peut donc être modélisé par une matrice de n lignes et 2 colonnes telle que la i-ième ligne $\begin{pmatrix} x_i & y_i \end{pmatrix}$ contienne les coordonnées du i-ième sommet $v_i = (x_i, y_i)$ pour tout $i \in [\![1, n]\!]$.

Pour déclarer une matrice en Python, on peut utiliser la commande `array` de la bibliothèque `numpy`. Par exemple, les commandes suivantes modélisent l'ensemble des quatre sommets d'un rectangle.

```
import numpy as np
V=np.array([[0,0],[0,1],[3,1],[3,0]])
```

On rappelle qu'on accède au nombre de lignes de la matrice `V` par la commande `len(V)` et à la i-ième ligne par `V[i-1]`. De plus, `[x,y]=V[i-1]` déclare deux variables `x` et `y` égales aux deux coefficients de la i-ième ligne de `V` (qu'on aurait aussi pu obtenir par `x=V[i-1][0]` et `y=V[i-1][1]`).

Exercice 2

Écrire une fonction `affiche_Sommets` *qui prend en argument une matrice* `V` *modélisant un ensemble de sommets puis qui place ces sommets dans un graphique (le choix du style d'affichage est libre). Quelle face de dé obtient-on avec les commandes suivantes ?*

```
V=np.array([[3,3],[1,3],[1,1],[3,1],[2,2]])
affiche_Sommets(V)
```

Pour déclarer une matrice de n lignes et 2 colonnes dont tous les coefficients sont nuls, on peut utiliser la commande `zeros([n,2])` de la bibliothèque `numpy`.

Pour choisir un nombre réel au hasard entre 0 et 1 en Python, on peut utiliser la commande `random` de la bibliothèque `random`.

```
import random as rd
rd.random()
```

Exercice 3

Écrire une fonction `Sommets_aleatoires` *qui prend en argument un entier* $n \geqslant 1$ *puis qui renvoie une matrice* `V` *modélisant un ensemble de* n *sommets dont chaque coordonnée est un nombre réel choisi au hasard entre* 0 *et* 1.

On pourra tester la fonction `Sommets_aleatoires` à l'aide de la fonction `affiche_Sommets`.

2.3 Modélisation des arêtes

Étant donné un ensemble de $n \geqslant 1$ sommets $V = \{v_1, v_2, \ldots, v_n\}$, chaque arête possible est de la forme $e = \{v_i, v_j\}$ avec $1 \leqslant i < j \leqslant n$. Le couple d'indices (i, j) des extrémités de l'arête peut donc être vu comme le couple d'indices d'un coefficient situé au-dessus de la diagonale d'une matrice carrée d'ordre n.

L'ensemble des arêtes E d'un graphe ayant V pour ensemble de sommets peut donc être modélisé par une matrice triangulaire supérieure d'ordre n dont les seuls coefficients non nuls (par exemple égaux à 1) sont ceux d'indices (i, j) tels que $i < j$ et que les sommets v_i et v_j sont voisins.

Par exemple, les commandes suivantes modélisent l'ensemble des quatre sommets d'un rectangle et l'ensemble de ses quatre arêtes.

```
V=np.array([[0,0],[0,1],[3,1],[3,0]])
E=np.array([[0,1,0,1],[0,0,1,0],[0,0,0,1],[0,0,0,0]])
```

Exercice 4

Écrire une fonction `nombre_Aretes` *qui prend en argument une matrice* `E` *modélisant un ensemble d'arêtes puis qui renvoie le nombre d'arêtes.*

Exercice 5

Écrire une fonction `affiche_Graphe` *qui prend en argument une matrice* `V` *modélisant un ensemble de sommets et une matrice* `E` *modélisant un ensemble d'arêtes puis qui place ces sommets et qui trace les segments correspondant à ces arêtes dans un même graphique (le choix du style d'affichage est libre). Quelle lettre obtient-on avec les commandes suivantes ?*

```
V=np.array([[0,2],[3,-1],[2,2],[1,5],[-1,-1]])
E=np.array([[0,0,1,1,1],[0,0,1,0,0],[0,0,0,1,0],[0,0,0,0,0],[0,0,0,0,0]])
affiche_Graphe(V,E)
```

Pour choisir un nombre entier au hasard entre deux nombres entiers a et b (inclus) en Python, on peut utiliser la commande `randint(a,b)` de la bibliothèque `random`. Par exemple, la commande suivante simule le résultat du lancer d'un dé équilibré à six faces.

```
rd.randint(1,6)
```

Exercice 6

Écrire une fonction `Aretes_aleatoires` *qui prend en argument un entier $n \geqslant 1$ puis qui renvoie une matrice* `E` *modélisant un ensemble d'arêtes d'un graphe ayant n sommets. Pour chaque arête possible, la fonction décidera au hasard (avec probabilité $1/2$) si elle fait partie ou non du graphe.*

Exercice 7

Écrire une fonction `Aretes_aleatoires_nbfixe` *qui prend en argument un entier $n \geqslant 1$ et un entier $p \geqslant 0$ puis qui renvoie une matrice* `E` *modélisant un ensemble d'exactement p arêtes d'un graphe ayant n sommets. La fonction choisira au hasard chacune des p arêtes parmi les arêtes possibles.*

On pourra tester les fonctions `Aretes_aleatoires` et `Aretes_aleatoires_nbfixe` à l'aide des fonctions `Sommets_aleatoires` et `affiche_Graphe`.

2.4 Quelques algorithmes

Exercice 8

Écrire une fonction `nombre_Voisins` *qui prend en argument une matrice* `V` *modélisant un ensemble de sommets $\{v_1, v_2, \ldots, v_n\}$ et une matrice* `E` *modélisant un ensemble d'arêtes puis qui renvoie une liste* `N` *dont le coefficient* `N[i-1]` *est égal aux nombres de voisins de v_i dans le graphe $G = (V, E)$.*

Exercice 9

Écrire une fonction `est_sans_Sommet_isole` *qui prend en argument une matrice* `V` *modélisant un ensemble de sommets et une matrice* `E` *modélisant un ensemble d'arêtes puis qui renvoie* `True` *si le graphe $G = (V, E)$ est sans sommet isolé, et* `False` *sinon.*

Exercice 10

Écrire une fonction `affiche_Chemin` *qui prend en argument une matrice* `C` *modélisant une suite de sommets $(v_0, v_1, v_2, \ldots, v_m)$ puis qui représente sur un graphique le chemin correspondant (le choix du style d'affichage est libre).*

Exercice 11

Écrire une fonction `distances_Sommets` *qui prend en argument une matrice* `V` *modélisant un ensemble de sommets $\{v_1, v_2, \ldots, v_n\}$ puis qui renvoie une matrice triangulaire supérieure* `D` *dont le coefficient* `D[i-1][j-1]` *est égal à la distance entre v_i et v_j pour tout $1 \leqslant i < j \leqslant n$ (ainsi la matrice* `D` *modélise l'ensemble de toutes les arêtes possibles).*

Exercice 12

Écrire une fonction `longueurs_Aretes` *qui prend en argument une matrice* `V` *modélisant un ensemble de sommets et une matrice* `E` *modélisant un ensemble d'arêtes puis qui renvoie une matrice* `L` *modélisant le même ensemble d'arêtes mais dont les coefficients non nuls sont égaux aux longueurs des arêtes correspondantes.*

Exercice 13

Écrire une fonction `longueur_totale` *qui prend en argument une matrice* `V` *modélisant un ensemble de sommets et une matrice* `E` *modélisant un ensemble d'arêtes puis qui renvoie la longueur totale du graphe* $G = (V, E)$.

3 Des graphes particuliers

Dans tout ce chapitre, on fixe un ensemble de sommets $V = \{v_1, v_2, \ldots, v_n\}$ dans le plan euclidien $\mathbb{R}^2$. Chacun des paragraphes suivants introduit une façon différente de relier ces sommets par des arêtes, définissant ainsi une suite de graphes particuliers.

3.1 NNG : le graphe du plus proche voisin

Définition 14

Le ***graphe du plus proche voisin*** *(*Nearest Neighbor Graph*) ayant* V *pour ensemble de sommets, est le graphe noté* $NNG(V) = (V, E)$ *dont chaque sommet est relié par une arête au sommet le plus proche. Plus précisément :*

$$E = \Big\{\{v, v'\} \mid v \in V \textit{ et } \mathrm{d}(v, v') = \min\Big\{\mathrm{d}(v, w) \,|\, w \in V\Big\}\Big\}.$$

Dans les rares cas où il existe plusieurs sommets plus proches d'un même sommet, le graphe du plus proche voisin n'est pas défini de manière unique.

Par définition, le graphe du plus proche voisin est planaire et sans sommet isolé. Mais il n'est en général pas connexe. D'autre part, la suite des longueurs des arêtes successives le long d'un chemin de $NNG(V)$ est nécessairement monotone. Par conséquent, le graphe du plus proche voisin ne contient aucun cycle.

Exercice 14

Écrire une fonction `NNG` *qui prend en argument une matrice* `V` *modélisant un ensemble de sommets puis qui renvoie une matrice* `E` *modélisant l'ensemble des arêtes du graphe du plus proche voisin. En déduire une fonction* `affiche_NNG` *qui permet de représenter* $NNG(V)$ *sur un graphique.*

3.2 MST : l'arbre couvrant minimal

Définition 15

*L'****arbre couvrant minimal*** *(*Minimal Spanning Tree*) ayant* V *pour ensemble de sommets, est l'arbre noté* $MST(V) = (V, E)$ *de plus petite longueur totale possible. Plus précisément :*

$$\sum_{\{v,v'\}\in E} \mathrm{d}(v, v') = \min\left\{\sum_{\{v,v'\}\in E'} \mathrm{d}(v, v') \mid (V, E') \textit{ est un arbre}\right\}.$$

L'arbre couvrant minimal n'est en général pas défini de manière unique (par exemple si V est constitué des sommets d'un polygone régulier). Mais on peut montrer que si toutes les distances entre deux sommets sont différentes alors il est unique.

Il existe de nombreux algorithmes de construction de l'arbre couvrant minimal. Par exemple, l'algorithme de Prim consiste à faire «croître» l'arbre depuis un sommet initial en ajoutant les arêtes les unes après les autres. À chaque étape, on ajoute la plus petite arête possible ayant exactement une extrémité dans l'arbre en construction (et donc l'autre extrémité parmi les sommets non encore «utilisés», afin de ne pas créer de cycles). En raisonnant par l'absurde, on peut démontrer que l'arbre obtenu après avoir utilisé tous les sommets a une longueur totale plus petite que n'importe quel autre arbre ayant les mêmes sommets.

Exercice 15

Écrire une fonction `MST` *qui prend en argument une matrice* `V` *modélisant un ensemble de sommets puis qui renvoie une matrice* `E` *modélisant l'arbre couvrant minimal (à l'aide de l'algorithme de Prim). En déduire une fonction* `affiche_MST` *qui permet de représenter $MST(V)$ sur un graphique.*

3.3 RNG : le graphe de voisinage relatif

Définition 16

Le ***graphe de voisinage relatif*** *(Relative Neighborhood Graph) ayant V pour ensemble de sommets, est le graphe noté $RNG(V) = (V, E)$ dont chaque couple de sommets est relié par un arête si et seulement s'il n'existe pas un troisième sommet plus proche des deux sommets du couple qu'ils ne le sont entre eux. Plus précisément :*

$$\{v, v'\} \in E \iff \forall w \in V \setminus \{v, v'\}, \max\Big\{\mathrm{d}(v, w), \mathrm{d}(v', w)\Big\} \geqslant \mathrm{d}(v, v').$$

Le graphe de voisinage relatif est défini de manière unique.

À l'exception des rares cas où trois sommets sont équidistants deux à deux (formant un triangle équilatéral), le graphe de voisinage relatif ne contient aucun triangle (puisque les sommets du côté le plus long ne sont pas reliés par une arête). Mais il contient en général des cycles.

Exercice 16

Écrire une fonction `RNG` *qui prend en argument une matrice* `V` *modélisant un ensemble de sommets puis qui renvoie une matrice* `E` *modélisant le graphe de voisinage relatif. En déduire une fonction* `affiche_RNG` *qui permet de représenter $RNG(V)$ sur un graphique.*

3.4 GG : le graphe de Gabriel

Définition 17

Le ***graphe de Gabriel*** *(Gabriel Graph) ayant V pour ensemble de sommets, est le graphe noté $GG(V) = (V, E)$ dont chaque couple de sommets est relié par un arête si et seulement si le disque dont le segment reliant les deux sommets du couple est un diamètre ne contient aucun autre sommet. Plus précisément :*

$$\{v, v'\} \in E \iff \forall w \in V \setminus \{v, v'\}, \mathrm{d}\Big(\tfrac{v+v'}{2}, w\Big) > \tfrac{\mathrm{d}(v,v')}{2}.$$

Ce graphe a été nommé en l'honneur de Kuno Ruben Gabriel (1929-2003), statisticien cosmopolite (né en Allemagne, émigré en France, étudié et enseigné en Israël, puis enseigné et mort aux États-Unis) qui l'a introduit en 1969.

Le graphe de Gabriel est défini de manière unique. Il contient en général des cycles, dont des triangles.

Exercice 17

Écrire une fonction `appartient_Disque_diametre` *qui prend en argument trois matrices* `v1`, `v2` *et* `w` *modélisant trois points v_1, v_2 et w du plan euclidien $\mathbb{R}^2$ puis qui renvoie* `True` *si w appartient au disque de diamètre $[v_1, v_2]$, et* `False` *sinon.*

Exercice 18

Écrire une fonction `GG` *qui prend en argument une matrice* `V` *modélisant un ensemble de sommets puis qui renvoie une matrice* `E` *modélisant le graphe de Gabriel. En déduire une fonction* `affiche_GG` *qui permet de représenter $GG(V)$ sur un graphique.*

3.5 DT : la triangulation de Delaunay

Définition 18

La ***triangulation de Delaunay*** *(Delaunay Triangulation) ayant V pour ensemble de sommets, est le graphe noté $DT(V) = (V, E)$ dont chaque triplet de sommets est relié par trois arêtes (formant ainsi un triangle) si et seulement si leur disque circonscrit ne contient aucun autre sommet. Plus précisément :*

$$\big(\{v, v'\}, \{v', v''\}, \{v'', v\}\big) \in E^3 \iff \forall w \in V \setminus \{v, v', v''\},\ w \notin D_{v,v',v''}$$

où $D_{v,v',v''}$ désigne le disque circonscrit au triangle de sommets v, v' et v''.

Ce graphe a été nommé en l'honneur du mathématicien russe Boris Delaunay (1790-1980) qui l'a introduit en 1924.

La triangulation de Delaunay n'est en général pas défini de manière unique (par exemple si V contient au moins quatre points cocycliques).

On peut montrer que la triangulation de Delaunay est bien une triangulation. Il s'agit même de la triangulation qui maximise le plus petit angle de tous les angles des faces triangulaires, évitant ainsi les triangles «allongés».

Exercice 19

Écrire une fonction `appartient_Disque_circonscrit` *qui prend en argument quatre matrices* `v1`*,* `v2`*,* `v3` *et* `w` *modélisant quatre points v_1, v_2, v_3 et w du plan euclidien $\mathbb{R}^2$ puis qui renvoie* `True` *si w appartient au disque circonscrit au triangle de sommets v_1, v_2 et v_3, et* `False` *sinon.*

Exercice 20

Écrire une fonction `DT` *qui prend en argument une matrice* `V` *modélisant un ensemble de sommets puis qui renvoie une matrice* `E` *modélisant la triangulation de Delaunay. En déduire une fonction* `affiche_DT` *qui permet de représenter $DT(V)$ sur un graphique.*

3.6 La hiérarchie de Toussaint

On peut démontrer que les graphes précédents sont successivement contenus les uns dans les autres, formant une suite de sous-graphes appelés la **hiérarchie de Toussaint**.

Théorème 4

On a pour tout ensemble de sommets $V \subset \mathbb{R}^2$:

$$NNG(V) \subset MST(V) \subset RNG(V) \subset GG(V) \subset DT(V).$$

On peut montrer que tous ces graphes sont planaires. Puisque $NNG(V)$ est sans point isolé, les autres graphes de la hiérarchie aussi. De même, puisque $MST(V)$ est connexe, les graphes suivants aussi. Les premiers cycles apparaissent dans $RNG(V)$, les premiers triangles dans $GG(V)$. Enfin, seul $DT(V)$ a toutes ses faces triangulaires.

Des exemples sont représentés sur la page suivante (à l'aide des fonctions en Python écrites dans les exercices). La colonne de gauche représente 50 sommets pris au hasard (premier graphique en haut), puis les différentes façons de relier ces sommets par des arêtes hiérarchie de Toussaint (de haut en bas). De même avec 100 sommets dans la colonne au centre, et 200 sommets dans la colonne de droite.