SNT — Localisation, cartographie et mobilité

Partie L1 — Se situer grâce aux satellites

Histoire, fonctionnement du GPS et trilatération

Ton téléphone sait où il se trouve, à quelques mètres près, n'importe où sur Terre. Comment est-ce possible ? Et à quoi cela sert-il ?

Vidéo · Repères historiques

- ▶ Découvrons l'histoire de la localisation : https://www.youtube.com/watch?v=XDsDVqCbBKk

Cours · Comment fonctionne le GPS ?

- ▶ KEZAKO : comment fonctionne un GPS ? : https://www.youtube.com/watch?v=WoqpQbWdacQ
- ▶ Galileo : le GPS européen : https://www.youtube.com/watch?v=e79tSIpLiDk

Chaque satellite émet en permanence un signal qui contient **l'heure exacte d'émission**, donnée par une horloge atomique très précise. Le récepteur (ton téléphone) note l'heure d'arrivée du signal : il en déduit la **durée du trajet**, puis la **distance** qui le sépare du satellite.

$$d = v \times t$$

où v est la vitesse du signal (c'est une onde radio, elle va à la vitesse de la lumière : environ $300\ 000$ km/s) et t la durée du trajet en secondes.

À retenir : la trilatération

Connaître sa distance à un satellite, c'est savoir qu'on est **sur une sphère** centrée sur ce satellite.

Avec **trois** satellites, les trois sphères se coupent en un point : c'est la position du récepteur. Un **quatrième** satellite sert à corriger l'horloge du récepteur, beaucoup moins précise que celles des satellites.

Cette méthode, fondée sur des **distances**, s'appelle la **trilatération**.

Vigilance

On entend souvent « triangulation » : ce mot désigne une méthode fondée sur des **angles**. Le GPS mesure des **distances** : c'est une **trilatération**.

Le GPS ne fait que **recevoir** des signaux : ce n'est pas lui qui transmet ta position. Ce sont les applications de ton téléphone qui peuvent l'envoyer sur Internet.

Exercices · Des durées aux distances

Exercice 2 — Le principe en quelques lignes 🌶

En t'aidant des deux vidéos, rédige dans ton traitement de texte un court paragraphe (5 à 8 lignes) qui explique le fonctionnement du GPS et de Galileo. Ajoute un schéma : tu peux le dessiner à la main et le photographier.

Réponses dans ton compte rendu (exercice 2)

Exercice 4 — À quelle distance est le satellite ? 🌶🌶

Un satellite émet un signal à 13 h 20 min 30,00 s et le smartphone le reçoit à 13 h 20 min 30,07 s.

1. Quelle est la durée du trajet, en secondes ?
2. Calcule la distance entre le satellite et le smartphone, en kilomètres.
3. Compare avec l'altitude des satellites GPS. Est-ce cohérent ?

Réponses dans ton compte rendu (exercice 4)

Activité · La trilatération sur une carte

Sur une carte des États-Unis, quatre « satellites » sont placés. Pour chaque personne, on connaît sa distance (en cm sur la carte) à chacun des satellites. On trace un **cercle** autour de chaque satellite : la personne est à l'intersection des cercles.

- ⬇ Carte GeoGebra (Base.ggb) : https://maths.kergot.com/informatique/SNT/Base.ggb
- ⬇ Les quatre cartes (zip) : https://maths.kergot.com/informatique/SNT/Localisation_TP1_Cartes.zip

1. Ouvre le fichier dans GeoGebra.
2. Choisis l'outil **Cercle (centre-rayon)**.
3. Clique sur un satellite, puis tape le rayon (la distance du tableau).

Nom	Satellite 1	Satellite 2	Satellite 3	Satellite 4	état
Jean	6,4	14,1	11,2	8,9	
Sophie	13	8	10	6,8	
Mathilde	3,9	11,3	7	4,5	
Pierre	7,8	9	10,6	4,9	

Exercice 7 — Qui est où ? 🌶🌶

Trouve dans quel État se trouve chaque personne. Fais une capture d'écran de chaque carte pour ton compte rendu.

Réponses dans ton compte rendu (exercice 7)

À retenir · Synthèse

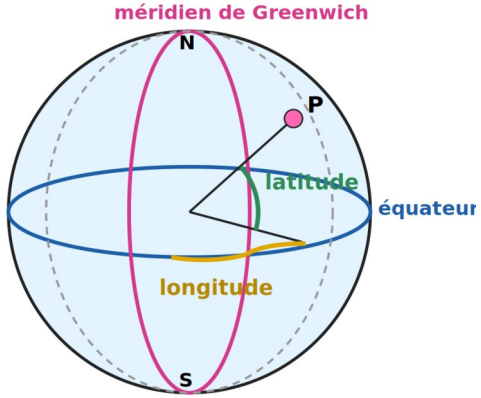
Ce que je sais expliquer

- Le récepteur mesure une **durée** et en déduit une **distance** : $d = v \times t$.
- Trois distances donnent la position (**trilatération**) ; le 4e satellite corrige l'horloge.
- Une erreur minuscule sur le temps donne une grande erreur de position.
- La géolocalisation a de nombreux usages, et pose des questions de vie privée.

Cours · Latitude et longitude

Tout point de la surface de la Terre est repéré par :

- sa **latitude** : l'angle qui le sépare de l'**équateur**, vers le **Nord** ou le **Sud**, de 0° à 90° ;
- sa **longitude** : l'angle qui le sépare du **méridien de Greenwich**, vers l'**Est** ou l'**Ouest**, de 0° à 180° ;
- et, si besoin, son **altitude** (hauteur par rapport au niveau de la mer).



Latitude (mesurée depuis l'équateur) et longitude (mesurée depuis le méridien de Greenwich).

À retenir : la convention des signes

On écrit toujours **la latitude d'abord, la longitude ensuite**.

En degrés décimaux, les directions deviennent des **signes** : Nord et Est sont **positifs**, Sud et Ouest sont **négatifs**.

Exemple : Santiago du Chili, 33,4569° S et 70,6483° O, s'écrit $(-33,4569\,;-70,6483)$.

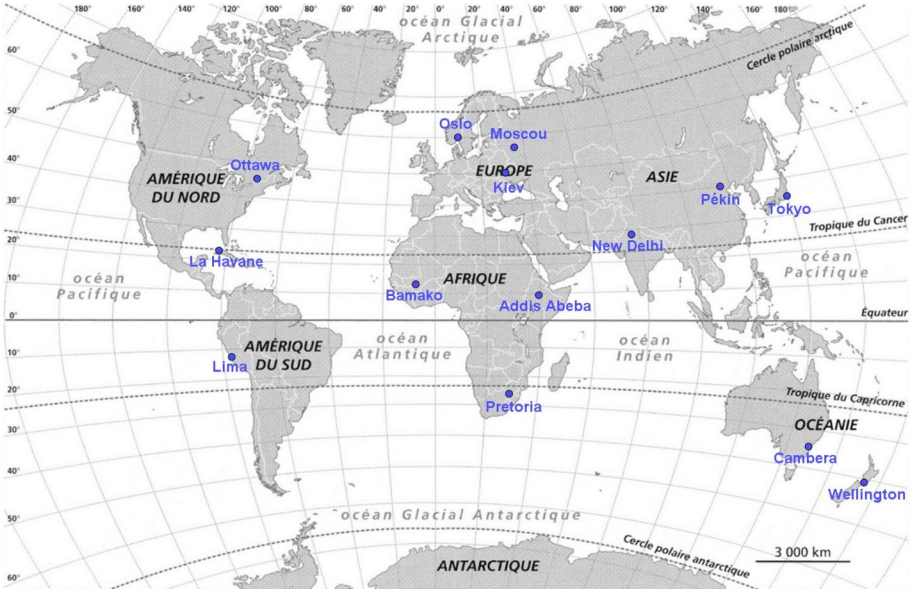
Partie L2 — Coordonnées géographiques

Latitude, longitude, et les différentes façons de les écrire

Exercices · Lire une carte du monde

Exercice 10 — Capitales du monde

À l'aide de la carte, complète le tableau (latitudes et longitudes arrondies à l'unité).



Planisphère : quelques capitales.

Capitale	Pays	Latitude	Longitude
Pékin			
	Norvège		
		12° S	77° O
	Canada		
		26° S	28° E
Moscou			
		13° N	8° O
Wellington			
	Ukraine		
		9° N	39° E

Réponses dans ton compte rendu (exercice 10)

Cours · Trois écritures d'une même position

Exemple : Bagnols-sur-Cèze.

Écriture	Exemple	Remarque
DMS : degrés, minutes, secondes	44° 09′ 35″ N ; 4° 37′ 03″ E	cartes papier, métadonnées des photos
DM : degrés, minutes décimales	44° 9,583′ N ; 4° 37,05′ E	GPS de navigation, bateaux
DD : degrés décimaux	44,1597° ; 4,6175°	cartes en ligne, Folium ; en général 4 à 6 décimales

À retenir : les conversions

1° = 60′ et **1′ = 60″**, donc **1° = 3600″**.

$$\mathrm{DD} = d + \frac{m}{60} + \frac{s}{3600}$$

De DD vers DMS : la partie entière donne les degrés ; on multiplie la partie décimale par 60 pour les minutes, puis encore par 60 pour les secondes.

Tu as programmé ces conversions en Python : partie 3, exercices 31 et 32.

Vigilance

Le signe s'applique à **tout** le nombre : 33° 27′ S donne $-\left(33+\frac{27}{60}\right)$, et non $-33+\frac{27}{60}$.

Ton convertisseur Python ne gère que des angles positifs : convertis la valeur sans signe, puis ajoute le signe (ou la lettre) à la fin.

Exercices · Convertir

Exercice 11 — Santiago 🌶🌶

Les coordonnées de Santiago sont $(-33,4569\,;-70,6483)$.

1. Dans quel hémisphère (Nord ou Sud) et de quel côté de Greenwich (Est ou Ouest) se trouve cette ville ?
2. De quel pays est-ce la capitale ?
3. Convertis ces coordonnées en DMS, à la main pour la latitude, avec ton programme pour la longitude.

Réponses dans ton compte rendu (exercice 11)

Exercice 14 — Bagdad 🌶🌶🌶

Les coordonnées de Bagdad sont 33° 20,4348′ N et 44° 24,0528′ E.

1. De quel pays est-ce la capitale ?
2. Convertis ces coordonnées DM en DMS.

Réponses dans ton compte rendu (exercice 14)

Synthèse

- Latitude (N/S depuis l'équateur) d'abord, longitude (E/O depuis Greenwich) ensuite.
- En DD : Sud et Ouest sont négatifs.
- DMS, DM et DD sont trois écritures d'un même angle ; 1° = 60′ = 3600″.

Partie L3 — Enquête : où ces photos ont-elles été prises ?

Les métadonnées cachées dans les photos

Cours · Les métadonnées d'une photo

Un fichier photo ne contient pas seulement l'image : il contient aussi des **métadonnées** (format EXIF) : date et heure, modèle du téléphone, réglages… et, si la géolocalisation de l'appareil photo est activée, **les coordonnées GPS** du lieu de la prise de vue.

- ⬇ Les photos de l'enquête (zip) : https://maths.kergot.com/informatique/SNT/Localisation_TP2_Photos.zip

1. Télécharge l'archive et décompresse-la dans ton dossier SNT.
2. Fais un clic droit sur une photo → *Propriétés* → onglet *Détails*.
3. Dans la rubrique *GPS*, la latitude s'affiche sous la forme `42; 38; 24.5616` : ce sont les **degrés, minutes et secondes**.
4. Pour vérifier un lieu : sur cartes.gouv.fr, *Recherche avancée* → *Coordonnées* (en degrés sexagésimaux, avec N/S et E/O), ou sur ta carte Folium.

Vigilance : vie privée

Une photo publiée avec ses métadonnées peut révéler **où tu habites**, ton lycée, tes lieux habituels.

Certains réseaux sociaux effacent ces données, d'autres non. Le plus sûr : désactiver la localisation dans l'appareil photo, ou effacer les métadonnées avant de partager.

Exercices · Les huit photos

Exercice 16 — Photo d'une façade 🌶🌶



Fichier « 6. Facade.jpg »

1. Relève les coordonnées GPS de la photo (DMS), puis convertis-les en DD.
2. De quel village est-ce la mairie ? Dans quelle rue la photo a-t-elle été prise ?

Réponses dans ton compte rendu (exercice 16)

Exercice 19 — Photo d'un phare 🌶🌶



Fichier « 9. Phare.jpg »

1. Relève les coordonnées GPS de la photo (DMS), puis convertis-les en DD.
2. Quelle est la ville la plus proche de ce phare ?

Réponses dans ton compte rendu (exercice 19)

Exercice 22 — Photo d'un pont 🌶🌶🌶



Fichier « 12. Pont.jpg »

1. Relève les coordonnées GPS de la photo (DMS), puis convertis-les en DD.
2. Dans quel fleuve se jette cette rivière ?

Réponses dans ton compte rendu (exercice 22)

Exercice 24 — Protéger ses photos 🌶🌶

Tu veux publier une photo prise dans ton jardin.

1. Quelles informations une personne malveillante pourrait-elle tirer de ses métadonnées ?
2. Sous Windows, dans *Propriétés* → *Détails*, trouve le lien qui permet de supprimer ces informations. Comment s'appelle-t-il ?
3. Sur ton téléphone, où se règle la géolocalisation des photos ?

Réponses dans ton compte rendu (exercice 24)

Partie L4 — Explorer les cartes publiques

cartes.gouv.fr : échelles, cartes anciennes, mesures, itinéraires

cartes.gouv.fr est le site public de cartographie de l'IGN (il remplace Géoportail). Il réunit des centaines de cartes : plans actuels, photographies aériennes, cartes anciennes… et des outils de mesure.

Méthode · Prendre en main le site

1. Ouvre cartes.gouv.fr et clique sur *Accéder aux cartes*. Pour les cookies, choisis *Refuser tous les cookies* : le site fonctionne très bien sans.
2. La barre *Rechercher un lieu* trouve une adresse ou un monument.
3. Le bouton *Catalogue de cartes* → *Cartes de référence* permet d'ajouter des fonds : *Plan IGN*, *Photographies aériennes*, *Carte de l'État-major 1820-1866*, *Carte de Cassini*…
4. Le bouton *Afficher plus* ouvre les outils : mesurer une distance ou une surface, itinéraire, zone selon le temps de trajet, coordonnées du curseur, profil altimétrique.

> **Vigilance**
>
> Les fonds de carte s'empilent comme des calques : le dernier ajouté cache les autres. Dans *Ma sélection de cartes*, tu peux masquer un calque ou changer l'ordre.

Exercice 25 — Où est exactement la tour Eiffel ? 🌶

1. Recherche la tour Eiffel.
2. Avec l'outil *Afficher les coordonnées du curseur*, relève sa position en **degrés décimaux** (4 décimales).
3. Dans les réglages de l'outil, passe en **degrés sexagésimaux** (DMS). Compare avec ton convertisseur Python.

Réponses dans ton compte rendu (exercice 25)

Exercice 26 — Du monde entier à la rue 🌶️

1. Dézoome au maximum. Que représente alors la barre d'échelle (en bas de la carte) ?
2. Zoome au maximum sur la tour Eiffel. Que représente la barre d'échelle ?
3. Explique pourquoi une carte en ligne n'a pas **une** échelle, mais une échelle qui change.

Réponses dans ton compte rendu (exercice 26)

Exercice 27 — Voyage dans le temps : Paris 🌶️🌶️

1. Ajoute le fond *Carte de l'État-major 1820-1866* et centre la carte sur Paris. Quelle grande différence vois-tu avec le Paris d'aujourd'hui ?
2. Ajoute la *Carte de Cassini* (XVIIIe siècle). Qu'est-ce qui est représenté, qu'est-ce qui ne l'est pas ?

Réponses dans ton compte rendu (exercice 27)

Exercice 30 — Trois distances pour un même trajet 🌶️🌶️

On veut aller du **35 avenue de la Bourdonnais, 75007 Paris** à l'**allée Léon-Bourgeois, 75007 Paris**. Détermine la distance :

1. à vol d'oiseau (outil *Mesurer une distance*) ;
2. à pied (outil *Itinéraire*) ;
3. en voiture (outil *Itinéraire*).
4. Classe ces trois distances et explique pourquoi elles sont différentes.

Réponses dans ton compte rendu (exercice 30)

Synthèse

- Une carte en ligne superpose des **calques** (plans, photos, cartes anciennes, données).
- L'échelle change avec le zoom.
- Distance à vol d'oiseau, à pied, en voiture : trois réponses différentes au même problème.
- Les données géographiques publiques (IGN) sont libres d'accès.

Partie L5 — Mes cartes avec Folium

Fabriquer ses propres cartes interactives en Python

Méthode · Avant de commencer

Ce TP prolonge la passerelle Folium du cours Python : commence par y installer Folium et par faire les exercices 35 à 38 si ce n'est pas déjà fait.

rappel.py

```
import folium

carte = folium.Map(location=[44.1646, 4.6103], zoom_start=16)
folium.Marker([44.1646098, 4.6103368], popup="Lycée Albert
Einstein").add_to(carte)
carte.save("ma_carte.html")
```

Cours · OpenStreetMap, la carte que tout le monde peut modifier

Le fond de carte utilisé par défaut par Folium vient d'**OpenStreetMap** : une carte **collaborative**, dessinée par des centaines de milliers de bénévoles, un peu comme Wikipédia. Ses données sont **libres** : n'importe qui peut les réutiliser, à condition de citer la source.

Folium propose d'autres fonds, avec le paramètre `tiles` :

fonds.py

```
carte = folium.Map(location=[44.1646, 4.6103], zoom_start=16,
tiles="cartodbpositron")
```

Valeur de `tiles`	Rendu
`"OpenStreetMap"`	le fond par défaut, très détaillé
`"cartodbpositron"`	fond clair et épuré, idéal pour faire ressortir les marqueurs
`"cartodbdark_matter"`	fond sombre

Exercices · Explorer et personnaliser

Exercice 33 — Trois fonds de carte 🌶

Crée trois cartes du lycée avec les trois fonds du tableau, enregistrées sous trois noms différents. Lequel préfères-tu pour une carte de vacances ? Pourquoi ?

Fichier : NOM_Prenom_ex33.py

Exercice 35 — Des marqueurs plus parlants 🌶🌶

Voici un marqueur qui affiche un texte au survol de la souris (`tooltip`) et qui a une couleur :

marqueur.py

```
folium.Marker([44.1646098, 4.6103368],
              popup="Lycée Albert Einstein",
              tooltip="Clique-moi !",
              icon=folium.Icon(color="green", icon="info-
sign")).add_to(carte)
```

Crée une carte de ta ville avec au moins **quatre marqueurs** de couleurs différentes selon leur catégorie (sport, culture, commerce…), chacun avec un `tooltip` et un popup.

Fichier : NOM_Prenom_ex35.py

Exercice 37 — Tracer un trajet 🌶🌶🌶

La fonction `folium.PolyLine` trace une ligne qui relie plusieurs points, donnés dans une **liste de listes** :

trajet.py

```
trajet = [[44.1646, 4.6103], [44.1602, 4.6196], [44.1523,
4.6290]]
folium.PolyLine(trajet, color="red", weight=5).add_to(carte)
```

Trace le trajet d'un voyage (réel ou imaginé) passant par au moins **quatre villes**, avec un marqueur sur chaque étape.

Fichier : NOM_Prenom_ex37.py

> **Et ensuite ?**
>
> Tu es prêt pour l'évaluation **« Les vacances du prof »** : à partir de vraies photos de voyage, tu fabriqueras ta propre carte.

Partie EV — Évaluation : les vacances du prof

Retrouver des lieux de voyage à partir de photos, et en faire une carte

Ton professeur t'attribue une archive de photos de vacances. À toi de retrouver où elles ont été prises, et de fabriquer la carte du voyage.

Évaluation · Consignes

1. Crée un dossier `Holidays` dans ton dossier SNT.
2. Télécharge l'archive indiquée par ton professeur et décompresse-la dans ce dossier. Choisis **au moins quatre** de tes photos préférées.
3. Pour chaque photo, relève les coordonnées GPS (en DMS, dans les métadonnées) et convertis-les en DD **avec ton programme Python**.
4. Avec une recherche d'images (Google Images, Lens…), identifie ce que montre chaque photo si ce n'est pas évident.
5. Crée un programme Python nommé `NOM_Prenom_vacances.py` dans le dossier `Holidays`.
6. Dans ce programme :

- en commentaire au début : ton nom, ta classe, le numéro de l'archive et le pays du voyage ;
- une carte centrée sur ta photo préférée, avec le plus grand zoom qui laisse voir tous les marqueurs ;
- un marqueur par photo, placé en **DD**, avec un popup qui décrit la photo, sa localisation (ville, rue…) et un petit commentaire personnel ;
- après chaque marqueur, une ligne de commentaire avec les coordonnées **DMS** relevées sur la photo.

> **Règles**
>
> Autorisés : ton polycopié (papier et en ligne), tes programmes, un moteur de recherche.

> Interdit : communiquer avec tes voisins.
>
> **Cinq minutes avant la fin**, vérifie ton travail et envoie ton fichier `.py` sur Pronote (ou à l'adresse donnée par ton professeur).

exemple_de_debut.py

```
# DUPONT Léa - 2nde 6 - archive 12 - Japon
import folium

carte = folium.Map(location=[35.6600, 139.6996], zoom_start=6)

folium.Marker([35.660081, 139.699647],
              popup="Carrefour de Shibuya, Tokyo. Le plus
célèbre passage piéton du monde !").add_to(carte)
# DMS : 35° 39' 36.29" N ; 139° 41' 58.73" E

carte.save("vacances.html")
```

Ressources · Les archives

- Archive 1 : https://maths.kergot.com/informatique/SNT/eval-geoloc/pack_1.zip
- Archive 2 : https://maths.kergot.com/informatique/SNT/eval-geoloc/pack_2.zip
- Archive 3 : https://maths.kergot.com/informatique/SNT/eval-geoloc/pack_3.zip
- Archive 4 : https://maths.kergot.com/informatique/SNT/eval-geoloc/pack_4.zip
- Archive 5 : https://maths.kergot.com/informatique/SNT/eval-geoloc/pack_5.zip
- Archive 6 : https://maths.kergot.com/informatique/SNT/eval-geoloc/pack_6.zip
- Archive 7 : https://maths.kergot.com/informatique/SNT/eval-geoloc/pack_7.zip
- Archive 8 : https://maths.kergot.com/informatique/SNT/eval-geoloc/pack_8.zip
- Archive 9 : https://maths.kergot.com/informatique/SNT/eval-geoloc/pack_9.zip
- Archive 10 : https://maths.kergot.com/informatique/SNT/eval-geoloc/pack_10.zip
- Archive 11 : https://maths.kergot.com/informatique/SNT/eval-geoloc/pack_11.zip
- Archive 12 : https://maths.kergot.com/informatique/SNT/eval-geoloc/pack_12.zip
- Archive 13 : https://maths.kergot.com/informatique/SNT/eval-geoloc/pack_13.zip
- Archive 14 : https://maths.kergot.com/informatique/SNT/eval-geoloc/pack_14.zip
- Archive 15 : https://maths.kergot.com/informatique/SNT/eval-geoloc/pack_15.zip
- Archive 16 : https://maths.kergot.com/informatique/SNT/eval-geoloc/pack_16.zip
- Archive 17 : https://maths.kergot.com/informatique/SNT/eval-geoloc/pack_17.zip

Évaluation · Grille de notation

Critère	Points
Fichier bien nommé, dans le dossier `Holidays` ; commentaire d'en-tête complet	2
Coordonnées DMS relevées pour 4 photos (en commentaire)	4
Conversions en DD justes	4
Carte centrée sur une photo, zoom adapté à tous les marqueurs	2
4 marqueurs bien placés (ordre latitude / longitude, signes)	4
Popups : description, localisation, commentaire personnel	3
Programme qui s'exécute sans erreur	1
Total	**20**

Bonus possibles : couleurs de marqueurs, `tooltip`, trajet tracé avec `PolyLine`, plus de quatre photos.

Partie L6 — Les graphes

Modéliser des relations : sommets, arêtes, distances

Cours · Qu'est-ce qu'un graphe ?

Un **graphe** représente des relations entre des objets :

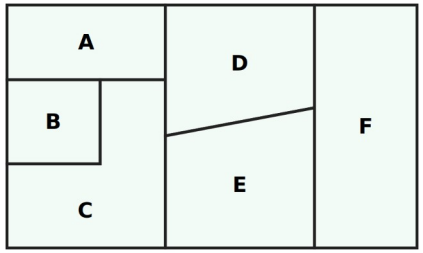
- les objets sont les **sommets** du graphe ;
- si deux objets sont en relation, leurs sommets sont reliés par une **arête**.

Réseau routier (villes et routes), réseau social (personnes et amitiés), plan de métro (stations et lignes) : ce sont des graphes.

Exercices · Modéliser

Exercice 39 — Le lotissement

Un lotissement est divisé en parcelles. Deux parcelles sont **voisines** si elles ont une frontière commune (pas seulement un coin).

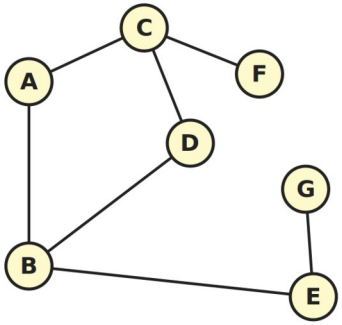


Le lotissement.

Dessine le graphe de voisinage : un sommet par parcelle, une arête entre deux parcelles voisines.

Réponses dans ton compte rendu (exercice 39)

Cours · Le vocabulaire des graphes



Le graphe d'exemple.

Mot	Définition	Exemple sur le graphe
adjacents	deux sommets reliés par une arête	A et B
chaîne	suite de sommets reliés deux à deux par des arêtes	C–A–B–E
longueur d'une chaîne	nombre d'arêtes de la chaîne	C–A–B–E a pour longueur 3
distance entre deux sommets	longueur de la plus **courte** chaîne qui les relie	distance de A à E : 2
excentricité d'un sommet	plus **grande** distance entre ce sommet et les autres	voir exercice 42
diamètre du graphe	plus grande des excentricités	
rayon du graphe	plus petite des excentricités	
centre du graphe	sommet(s) dont l'excentricité est égale au rayon	

Vigilance

La distance est la longueur de la chaîne la plus **courte** ; l'excentricité est la plus **grande** de ces distances. On mélange souvent les deux.

Exercices · Utiliser le vocabulaire

Exercice 41 — Des chaînes 🌶

1. Donne une autre chaîne que C–A–B–E qui relie C à E. Quelle est sa longueur ?
2. Donne une chaîne reliant F à G. Est-ce la plus courte ?
3. Quelle est la distance entre F et G ?

Réponses dans ton compte rendu (exercice 41)

Exercice 42 — Le tableau des distances 🌶🌶

Recopie et complète le tableau des distances entre les sommets du graphe d'exemple, puis la colonne des excentricités.

	A	B	C	D	E	F	G	Excentricité
A	0							
B		0						
C			0					
D				0				
E					0			
F						0		
G							0	

Déduis-en le diamètre, le rayon et le centre du graphe.

Réponses dans ton compte rendu (exercice 42)

Exercice 43 — Cas d'école : une famille 🌶🌶

Un réseau social est formé par une famille de 5 personnes : deux membres quelconques de la famille sont toujours en contact.

1. Trace le graphe correspondant (sommets A à E). Un tel graphe, où tous les sommets sont reliés deux à deux, est dit **complet**.
2. Donne le diamètre, le rayon et le centre de ce graphe.
3. Une personne de la famille (au choix) est aussi en contact avec quelqu'un d'extérieur (F), qui ne connaît qu'elle. Donne le diamètre, le rayon et le centre du nouveau graphe.

Réponses dans ton compte rendu (exercice 43)

Synthèse

- Un graphe : des **sommets** et des **arêtes**.
- Distance = longueur de la plus courte chaîne.
- Excentricité d'un sommet = sa plus grande distance aux autres.
- Diamètre = plus grande excentricité ; rayon = plus petite ; centre = sommets d'excentricité égale au rayon.

Partie L7 — Le plus court chemin

L'algorithme de Dijkstra, à la main, animé et en Python

Cours · Graphes pondérés

Dans un **graphe pondéré**, chaque arête porte un nombre, son **poids** : une distance en km, une durée en minutes, un prix… La longueur d'une chaîne est alors la **somme des poids** de ses arêtes.

Les GPS et les sites d'itinéraires cherchent la chaîne de plus petit poids entre deux sommets. L'algorithme le plus célèbre pour cela a été inventé par Edsger **Dijkstra** en 1956.

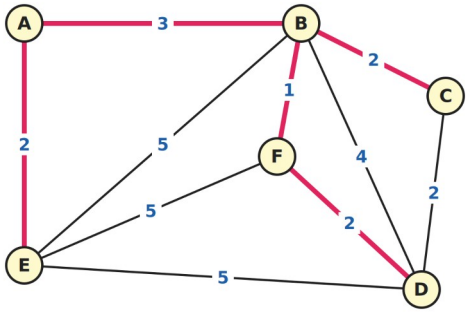
- ▶ Graphes : les bases : https://www.youtube.com/watch?v=6_3N0BbP89U
- ▶ Graphes : les bases (2) : https://www.youtube.com/watch?v=SKh5CLqBhUw

- ▶ Dijkstra 1 : https://www.youtube.com/watch?v=rHylCtXtdNs
- ▶ Dijkstra 2 : https://www.youtube.com/watch?v=MybdP4kice4
- ▶ Dijkstra 3 : https://www.youtube.com/watch?v=JPeCmKFrKio
- ▶ Dijkstra 4 : https://www.youtube.com/watch?v=HPPKiCzjX-Q

Méthode · L'algorithme de Dijkstra, version dessin

1. **Au départ :** écris 0 sur le sommet de départ et ∞ sur tous les autres.
2. **Choisis** le sommet non traité qui a la plus petite valeur : c'est le sommet **actif**. Entoure-le.
3. **Regarde ses voisins non traités.** Pour chacun, calcule : valeur du sommet actif + poids de l'arête.
4. Si ce nombre est **plus petit** que la valeur du voisin : remplace la valeur, **trace** en couleur l'arête qui vient du sommet actif, et **raye** l'ancienne arête colorée qui arrivait à ce voisin (s'il y en avait une).
5. Si ce nombre est plus grand ou égal : **ne change rien**.
6. Le sommet actif est maintenant **traité** : encadre sa valeur, elle ne changera plus.
7. **Recommence** à l'étape 2 jusqu'à ce que tous les sommets soient traités.

À la fin, la valeur de chaque sommet est sa distance au départ, et les arêtes colorées forment les plus courts chemins.



Graphe de la fiche méthode : distances depuis A.

Exemple : le graphe de la fiche méthode, depuis A

1. **Départ :** 0 sur A, ∞ sur tous les autres sommets.
2. **Sommet actif A.** B reçoit 3 et E reçoit 2 : on trace A–B et A–E.
3. **Sommet actif E** (car 2 < 3). F reçoit 2 + 5 = 7 et D reçoit 2 + 5 = 7 : on trace E –F et E–D. Vers B : 2 + 5 = 7, plus grand que 3, rien à faire.
4. **Sommet actif B** (3). Vers F : 3 + 1 = 4 < 7, on **raye** E–F et on trace B–F. Vers D : 3 + 4 = 7, égalité, rien à faire. Vers C : 3 + 2 = 5, on trace B–C.
5. **Sommet actif F** (4). Vers D : 4 + 2 = 6 < 7, on **raye** E–D et on trace F–D.
6. **Sommet actif C** (5). Vers D : 5 + 2 = 7, pas mieux que 6, rien à faire.
7. **Dernier sommet D** (6) : terminé.

Pour aller de A à D : **A–B–F–D**, longueur 6. Pour aller en C : **A–B–C**, longueur 5.

Vigilance

En cas d'égalité (par exemple B → D donne 7, comme E → D), on **ne change rien** : l'ancien chemin est aussi court.

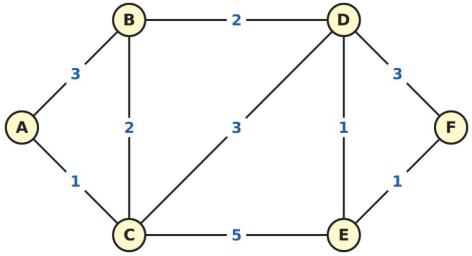
Une valeur encadrée (sommet traité) ne change plus jamais.

Voir l'algorithme en action

L'**animation Dijkstra** rejoue l'algorithme pas à pas sur les graphes du cours. En **mode pédagogique**, c'est toi qui choisis le sommet suivant et qui calcules les distances. Tu peux aussi y construire ton propre graphe.

Exercices · Chercher le plus court chemin

Exercice 45 — De A à F 🌶🌶



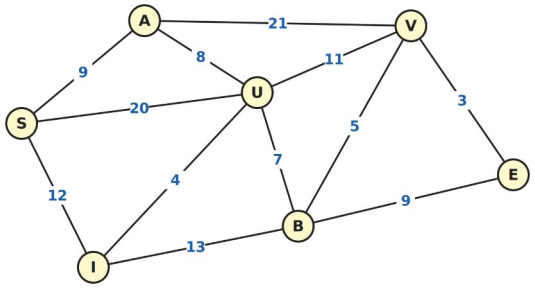
Les nombres sont des distances en kilomètres.

1. Applique l'algorithme de Dijkstra depuis A, en redessinant le graphe à chaque étape comme dans la méthode.
2. Quel est le plus court chemin de A à F ? Quelle est sa longueur ?

3. Vérifie avec l'animation (graphe « TP : de A à F »).

Réponses dans ton compte rendu (exercice 45)

Exercice 46 — Un dernier pour la route 🌶🌶🌶



Graphe à sept sommets.

1. Applique l'algorithme de Dijkstra depuis S, en redessinant le graphe à chaque étape.
2. Quel est le plus court chemin de S à E ? Sa longueur ?
3. Le chemin direct S–U fait 20. Pourquoi n'est-il pas utilisé ?

Réponses dans ton compte rendu (exercice 46)

Python · Des itinéraires réels

Le module **pyroutelib3** calcule un vrai itinéraire en téléchargeant les rues d'OpenStreetMap, puis en appliquant un algorithme de plus court chemin de la famille de Dijkstra.

1. Installe le module `pyroutelib3` comme tu as installé Folium.
2. Copie et exécute le programme ci-dessous (il faut une connexion Internet : le premier calcul peut prendre une minute).

itineraire.py

```
import folium
import pyroutelib3

# le réseau des chemins piétons, téléchargé au fur et à mesure
graphe =
pyroutelib3.osm.LiveGraph(pyroutelib3.osm.FootProfile())
depart = graphe.find_nearest_node((44.16213, 4.617122))
arrivee = graphe.find_nearest_node((44.163942, 4.611689))
chemin = pyroutelib3.find_route_without_turn_around(graphe,
depart.id, arrivee.id)
points = [graphe.get_node(n).position for n in chemin]

carte = folium.Map(location=[44.163, 4.614], zoom_start=16)
folium.PolyLine(points, color="red", weight=5).add_to(carte)
folium.Marker(points[0], popup="Départ").add_to(carte)
folium.Marker(points[-1], popup="Arrivée").add_to(carte)
carte.save("itineraire.html")
```

Exercice 48 — Mon itinéraire 🌶🌶

1. Exécute le programme et ouvre `itineraire.html`. Combien de points la variable `points` contient-elle ? (Ajoute `print(len(points))`.)
2. Modifie le programme pour tracer un itinéraire à pied de ton choix, par exemple de la gare au lycée.
3. Remplace `FootProfile()` par `CarProfile()`. Le trajet change-t-il ? Pourquoi ?
4. Compare la longueur de ton trajet à pied avec celle donnée par l'outil *Itinéraire* de cartes.gouv.fr.

Fichier : NOM_Prenom_ex48.py

Synthèse

- Dans un graphe pondéré, la longueur d'une chaîne est la somme des poids.
- Dijkstra : on fixe à chaque étape le sommet non traité le plus proche du départ, et on améliore ses voisins.
- Les GPS utilisent des algorithmes de la même famille sur le graphe des routes.