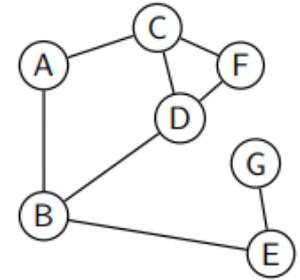


Etude n°1 : Notion de graphe

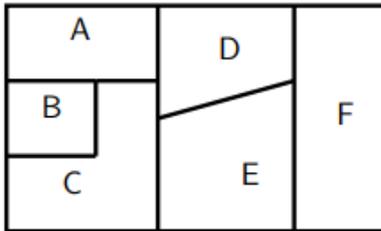
Un **graphe** permet de représenter des relations entre différents objets :

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

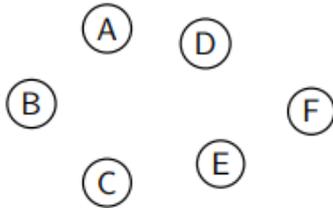


Exemple :

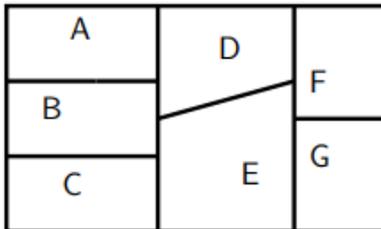
Un lotissement est divisé en parcelles :



Compléter le graphe représentant les relations de voisinage entre les parcelles.



Le propriétaire de la parcelle B achète un bout de la parcelle C, tandis que la parcelle F est séparée en deux. Représenter le nouveau graphe de voisinage.

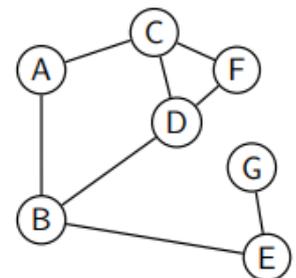


Etude n°2 : Vocabulaire des graphes

Définitions

1. Dans le graphe ci-contre :

- les **sommets** A et B sont reliés par une arête : ils sont **adjacents** ;
 - une **chaîne** reliant C à E est : C-A-B-E, de longueur 3 ;
 - une autre, de longueur 4 :
 - une chaîne de F à G :
- Est-ce la plus courte entre ces deux sommets ?



2. La **distance** entre deux sommets est la longueur de la plus petite chaîne les reliant.

Compléter les colonnes A à G du tableau des distances ci-contre.

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

3. Le **diamètre** d'un graphe est la distance maximale entre deux sommets du graphe.

Le diamètre du graphe ci-dessus est :

4. L'**excentricité** d'un sommet est la plus grande distance entre ce sommet et un autre sommet du graphe.

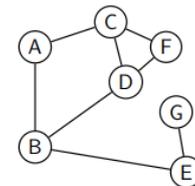
Compléter la colonne « Excentricité » du tableau du 2.

5. Le **rayon** d'un graphe est la plus petite excentricité de ses sommets.

Le rayon du graphe ci-dessus est :

6. Le **centre** d'un graphe est l'ensemble des points dont l'excentricité est égale au rayon du graphe.

Le centre du graphe ci-dessus est :



Cas d'école numéro 1

On considère le réseau social formé par une famille de 5 personnes : deux membres quelconques de la famille sont toujours en contact.

- Tracer un graphe correspondant à cette situation ; les sommets seront notés de A à E. Un tel graphe est dit **complet**.
- Préciser le diamètre, le rayon et le centre de ce graphe.
- Une personne de la famille (au choix) établit un lien exclusif avec quelqu'un d'extérieur (F). Déterminer le diamètre, le rayon et le centre de ce nouveau graphe.

Cas d'école numéro 2

Deux familles, l'une de 5 personnes (A à E), l'autre de 4 personnes (F à I), sont chacune représentées par des graphes complets (voir ci-dessus).

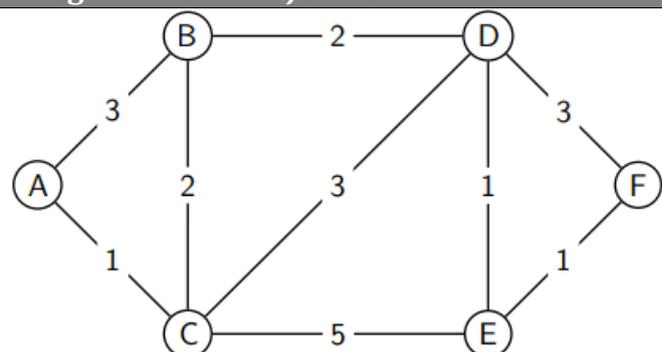
Elles sont en contact via deux de leurs membres (un dans chaque famille : A et F par exemple).

- Préciser le diamètre, le rayon et le centre de ce graphe.
- Déterminer ce que deviennent ces résultats si une nouvelle arête est tracée entre B et F.
- Même question, avec les deux familles mises en contact uniquement par un intermédiaire : J a un lien avec A et avec F.

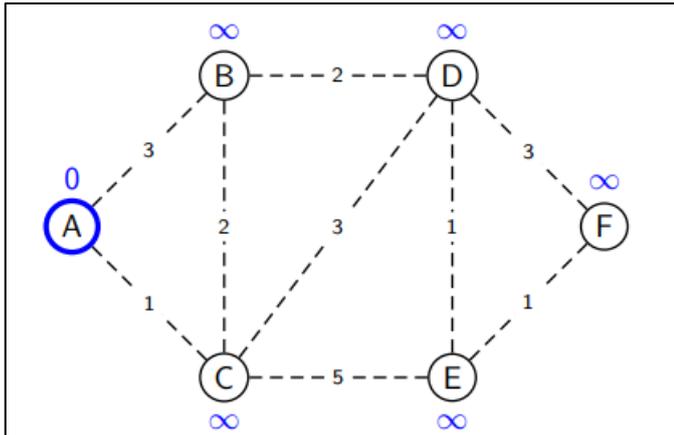
Etude n°3 : Comprendre l'algorithme de Dijkstra

Dans le graphe ci-contre, les sommets représentent des lieux (villes, etc.) et les nombres sur les arêtes des distances (en kilomètres, en minutes ou autre). Pour calculer des itinéraires (plus courts, plus rapides, etc.), de nombreux algorithmes s'appuyant sur la théorie des graphes ont été développés.

Dans cette partie, l'objectif est de calculer l'itinéraire le plus court reliant les sommets A et F.



L'algorithme de Dijkstra est un des algorithmes permettant de résoudre ce problème de plus court chemin. Il porte le nom de son inventeur, l'informaticien néerlandais Edsger Dijkstra, et a été publié en 1959.

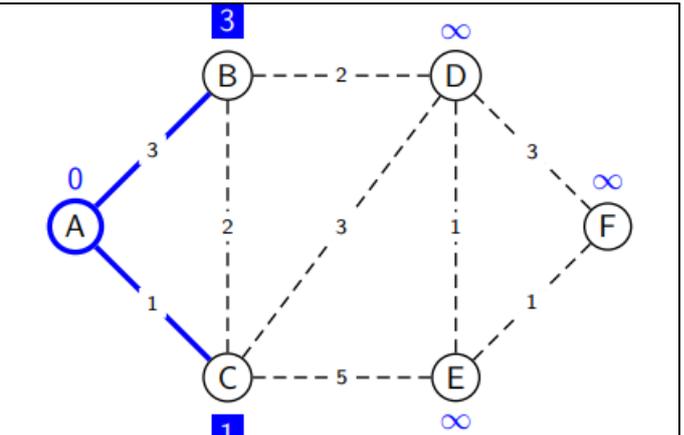


Initialisation

On se place sur le sommet A, et on « oublie » toutes les arêtes.

Chaque sommet est étiqueté par la distance la plus courte le reliant à A :

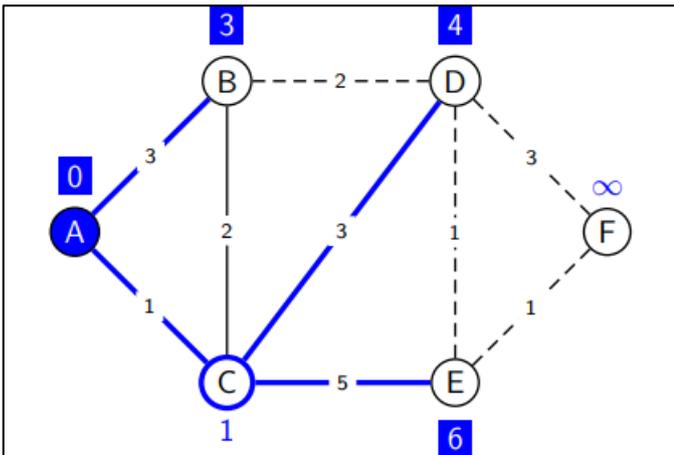
- la distance entre A et A est 0 ;
- la distance entre A et tout autre sommet est infinie.



Premier sommet : A

Les arêtes issues de A sont tracées. Les distances de A aux sommets B et C sont désormais connues :

- la distance entre A et B est 3 ;
- la distance entre A et C est 1 ;
- la distance entre A et A reste 0, et les autres restent infinies.

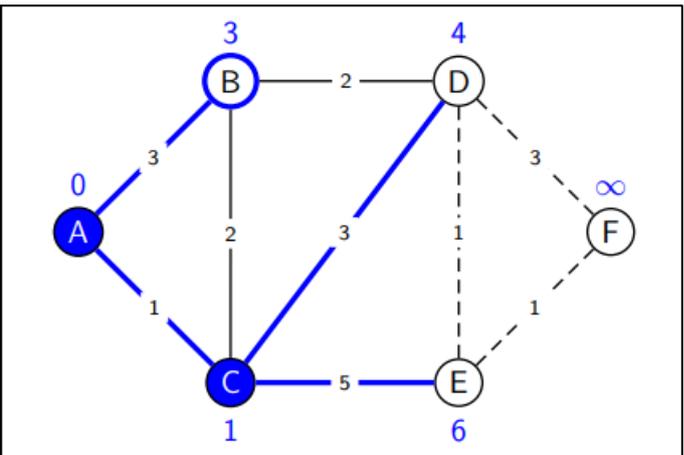


Sommet suivant : C

Le sommet sélectionné suivant est celui, non encore traité, qui porte la plus petite étiquette : le sommet C.

On considère les arêtes issues de C, et on actualise les distances entre A et chacun des sommets :

- Le chemin A-C-A est de longueur 2 : il est plus long que l'étiquette actuelle de A (0), qui ne change donc pas ;
- A-C-B est de longueur 3, identique à l'étiquette de B : celle-ci ne change pas ;
- A-C-D est de longueur 4, ce qui est mieux que ∞ ;
- A-C-E est de longueur 6, ce qui est mieux que ∞ .



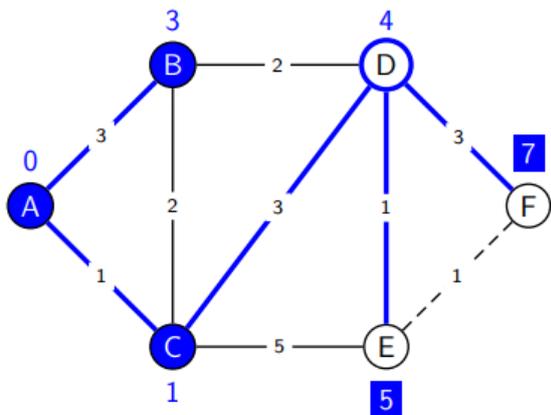
Sommet suivant : B

B est le sommet non encore traité portant la plus petite étiquette.

On considère les arêtes issues de B, et on actualise les distances entre A et chacun des sommets :

- A-B-A est de longueur $3 + 3 = 6$: trop long ;
- A-B-C est plus long que le chemin A-C déjà mémorisé, et ce chemin rallongerait aussi le trajet de A à E ;
- A-B-D

Conclusion : aucune étiquette ne change.

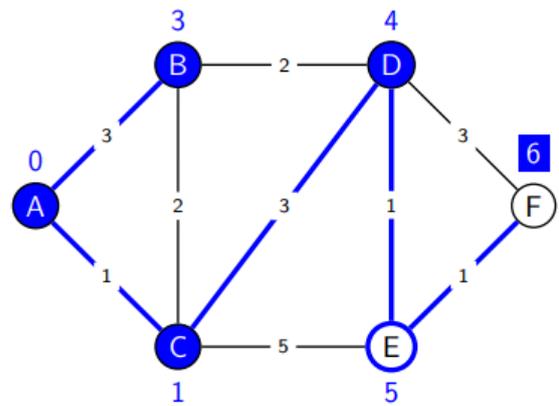


Sommet suivant : D

Même opération que précédemment :

- A-D-F
- A-D-E

Les chemins les plus courts depuis A sont actualisés et mémorisés (en bleu).



Dernier sommet : E

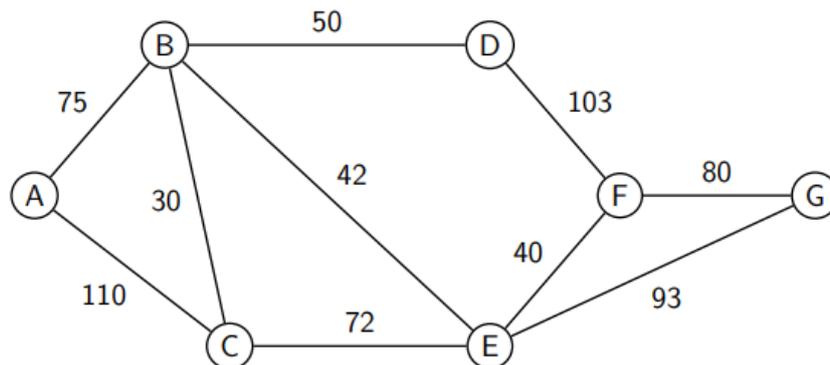
Même opération que précédemment :

- A-E-F
- A-E-C

Conclusion : le chemin le plus court de A à F est modifié, il devient ; l'arête peut être oubliée.

Exemple :

Le graphe ci-dessous représente le plan d'un centre de vacances. Les arêtes représentent les allées et les sommets, les carrefours. On a indiqué sur chaque arête la longueur en mètres des allées entre deux carrefours.



Déterminer, en utilisant l'algorithme de Dijkstra, le trajet le plus court pour aller du carrefour A au carrefour G.

Etude n°4 : Itinéraire avec Python

Ouvrir Python et installer la bibliothèque pyrouelib3. Saisir puis tester le programme ci-dessous :

```

from pyrouelib3 import Router
import folium
router = Router("foot")
depart = router.findNode(44.16213, 4.617122)
arrivee = router.findNode(44.163942, 4.611689)
status, route = router.doRoute(depart, arrivee)
if status == 'success':
    routeLatLons = list(map(router.nodeLatLon, route))
c= folium.Map(location=[44.16213, 4.617122],
zoom_start=15)
for coord in routeLatLons:
    coord=list(coord)
    folium.Marker(coord).add_to(c)
c.save('monitineraire.html')

```

Modifiez le programme pour qu'apparaisse sur une carte, un itinéraire de votre choix.