

## Séance 3 : Calculs d'itinéraires

**Sources** : Activité proposée par les formateurs académiques. Inspiré de SNT Delagrave pour l'algorithme naïf qui ne fonctionne pas.

**Thème** : Localisation, cartographie et mobilité

**Environnement envisagé** : PC pour première et dernière partie

**Prérequis** : aucun

**Durée** :

- séance 2h00

**Objectifs** :

- Représentation de cartes par un graphe
- Aborder le calcul d'itinéraires :
  - avec un logiciel,
  - à la main sur un petit graphe avec un algo naïf
  - avec Python et l'algorithme de Dijkstra (savoir modifier un programme donné)
  - à la main avec Dijkstra (vidéo d'explication fournie)

**Le but n'est pas de fournir des séquences clé en main, mais une base à retravailler et à adapter.**

# Séance 3 : Calculs d'itinéraires

## 1. Avec un logiciel

1. En utilisant Openstreetmap, déterminer le plus court chemin (en durée) pour aller de la gare de Sens jusqu'au lycée :
  - a) en vélo (copier ci-dessous la carte proposant l'itinéraire)

- b) en voiture (copier ci-dessous la carte proposant l'itinéraire)

Mais comment l'application a pu trouver ce chemin automatiquement ?

## 2. Algorithmes de calculs d'itinéraires

Sur papier, traduire une carte ou un tableau par un graphe pour avoir une représentation plus simple à manipuler.

On donne un algorithme simple de calcul d'itinéraire (mais pas le plus efficace) et on l'applique sur le graphe précédent.

Celui-ci fonctionne dans certains cas mais pas toujours => il faut un algo plus élaboré

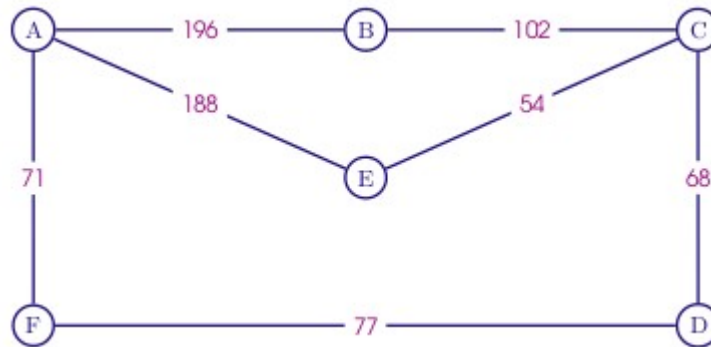
L'objectif des parties suivantes est de comprendre comment un logiciel parvient à calculer le plus court chemin entre deux points sur une carte.

Une société de transport propose certaines destinations. On donne ci-dessous les liaisons mises en place et la durée (en minute) de ces liaisons. Si aucune valeur n'est indiquée, c'est que la liaison n'existe pas entre les deux villes.

	A	B	C	D	E	F
A	0	196			188	71
B	196	0	102			
C		102	0	68	54	
D			68	0		77
E	188		54		0	
F	71			77		0

1. Représenter ce tableau par un graphe (Les sommets seront les villes et les arêtes, si elles existent, le temps de trajet).

Un graphe possible est



2. Un algorithme possible pour calculer le plus court chemin (en durée) est le suivant :

Se placer dans la ville de départ  
 Tant que la ville d'arrivée n'est pas atteinte :  
     depuis la ville courante, aller si possible dans la ville voisine (non déjà visitée)  
     la plus proche  
 Fin Tant que

- a) Appliquer l'algorithme au graphe obtenu précédemment pour calculer le plus court chemin entre A et D.  
 A – F – D de durée  $71+77 = 148$
- b) Cet algorithme donne-t-il le plus court chemin ? Si oui, le justifier, sinon donner un chemin plus court.  
 Oui car la première étape des autres chemins est de durée supérieure à 145
- c) Appliquer l'algorithme au graphe obtenu précédemment pour calculer le plus court chemin entre D et A.  
 D – C – E – A : durée  $68+54+188 = 310$
- d) Est-ce le plus court chemin ?  
 Non, le plus court chemin est celui trouvé précédemment. On aurait du avoir le même itinéraire. L'algo n'est pas fiable.

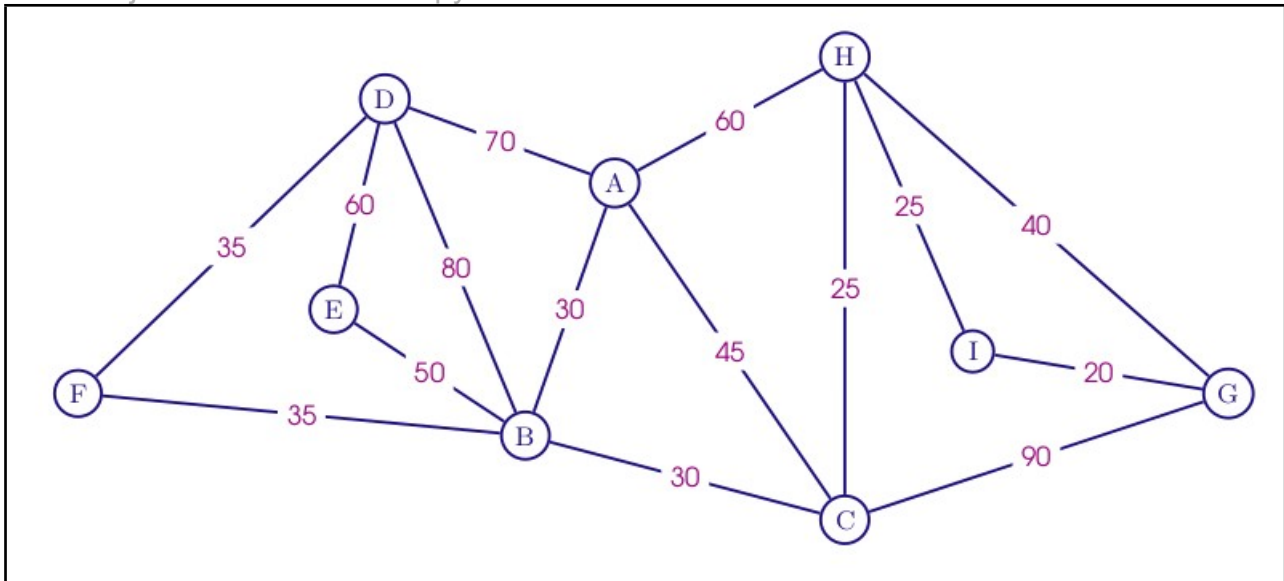
### 3. Algorithme de Dijkstra

L'algorithme précédent est simple mais ne fonctionne pas toujours. L'algorithme de Dijkstra est un algorithme plus complexe mais plus efficace qui est utilisé par les systèmes de cartographie. Le principe est de sélectionner à chaque étape le meilleur prédécesseur.

Le programme DijkstraExemple.py est un programme python permettant d'appliquer l'algorithme de Dijkstra au graphe ci-dessous.

**On ne demande pas de comprendre le programme mais juste de savoir modifier l'entrée des données. Seuls des éléments parmi les lignes 94 à 110 sont à modifier.**

1. Ouvrir le fichier DijkstraExemple.py. Il est prévu pour le graphe étudié précédemment. Exécuter le programme et vérifier que le résultat donné est bien le plus court chemin.
2. Modifier le programme (lignes 94 à 110) pour déterminer le plus court chemin du sommet G au sommet D sur le graphe ci-dessous. Voir fichier DijkstraCorrectionExos.py



3. La liaison entre C et H est coupée. Quel est alors le plus court chemin ? Attention il faut supprimer lien de C vers H et celui de H vers C
4. Sur youtube, visionner la vidéo concernant l'algorithme de Dijkstra de la chaine « 5 minutes pour comprendre ».  
<https://www.youtube.com/watch?v=MybdP4kice4>  
la vidéo est dans le répertoire
5. En suivant la démarche présentée dans la vidéo, compléter le tableau ci-dessous pour déterminer le plus court chemin entre les sommets G et D (graphe initial, comportant le chemin entre C et H).

G	A	B	C	D	E	F	H	I
0								
x			90(G)				40(G)	20(G)
x			90(G)				40(G)	x
x	100(H)		65(H)				x	x
x	100(H)	95(C)	x				x	x
x	100(H)	x	x	175(B)	145(B)	130(B)	x	x
x	x	x	x	170(A)	145(B)	130(B)	x	x
x	x	x	x	165(F)	145(B)	x	x	x

Le plus court chemin entre les sommets G et D est GHCBFD de longueur  $165 = 40+25+30+35+35$