

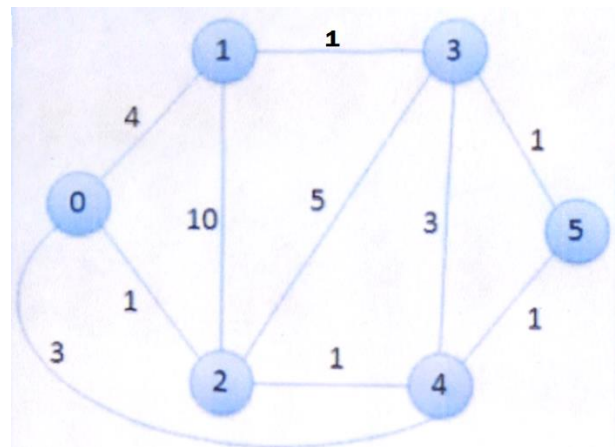
Cartographie -Algorithme de Dijkstra-

L'algorithme de Dijkstra est un algorithme très puissant permettant de trouver l'itinéraire le plus court entre un point et un autre.

Un réseau de villes et les routes qui les relient peuvent être représentés par un **graphe**. Le temps de trajet (par exemple) est alors indiqué sur l'**arête** qui est un lien entre deux villes.

I/ L'algorithme de Dijkstra

On considère le **graphe** suivant (à droite). Les **sommets** sont représentés ici par des nombres entiers successifs en partant de 0, l'ordre n'ayant pas d'importance.
Chaque **arête** (lien entre deux sommets) est pondérée par une valeur qui pourrait correspondre ici à la distance séparant deux villages.



Pour exploiter le graphe par l'algorithme de Dijkstra, on va créer la **matrice d'adjacence** qui va recenser toutes les liaisons possibles et en plus leur poids (distance entre deux villages ici).
Si une liaison directe n'est pas possible, on écrit zéro.

Vers	0	1	2	3	4	5
0	0	4	1	0	3	0
1		0				
2			0			
3				0		
4					0	
5						0

Pour aller de 0 à 1 : il existe un chemin direct et il mesure 4 km (d'où le « 4 »).

Pour aller de 0 à 2 : il existe un chemin direct et il mesure 1 km (d'où le « 1 »).

Pour aller de 0 à 3 : il n'existe pas de chemin direct (d'où le « 0 »).

1/ **Compléter** la matrice d'adjacence.

Appeler le professeur pour validation.

2/ **Copier-coller** le programme « Cartographie.Dijkstra.py » dans Jupyter, lien ici :

<https://github.com/Imayer65/SNT/blob/Private/Partie.III.Cartographie/Cartographie.Dijkstra.py>

Bien observer la donnée *adjacenceMatrice* et la **comparer** à celle de la question précédente.

```

52 # On passe le sommet de départ puis le sommet d'arrivée
53 algorithmDijkstra(adjacenceMatrice, 0, 5)

```

Le programme va chercher à déterminer le chemin le plus court entre le sommet « 0 » et le sommet « 5 »

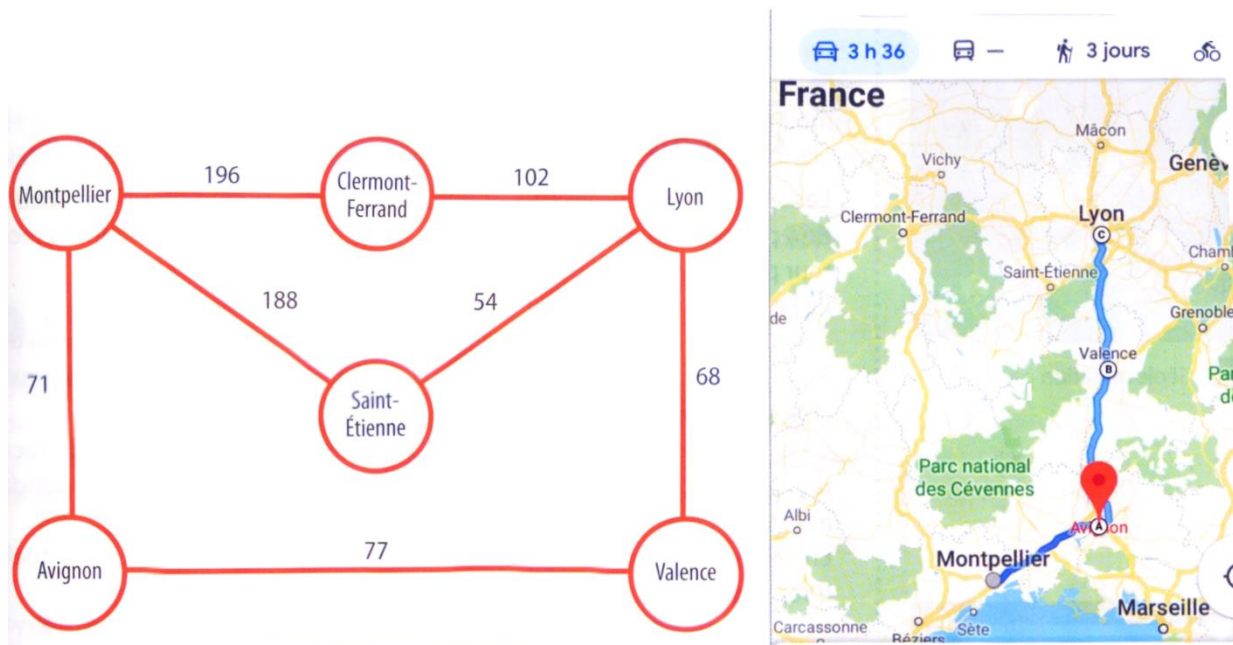
3/ **Exécuter** le programme et **vérifier** que l'on trouve 0 -> 2 -> 4 -> 5.

4/ **Déterminer** le chemin le plus court entre le sommet « 1 » et le sommet « 4 » en **modifiant** la ligne `algorithmDijkstra(adjacenceMatrice, 0, 5)` du programme précédent.

II/ Application à un réseau routier

Vérifier que le trajet proposé est bien le plus court à l'aide du programme précédent.

Aide : il faut établir la matrice d'adjacence de ce graphe avant !



Appeler le professeur pour validation