-	Ī	Ρ	1	7
٠	T	N	S	6

Algorithme de Dijkstra

L'objectif de cette activité est de découvrir un algorithme permettant de trouver le plus court chemin dans un graphe. Il sera illustré ici sur l'exemple de la recherche d'un itinéraire routier, mais est aussi utilisé dans le protocole de routage OSPF..

I - Découverte de l'algorithme

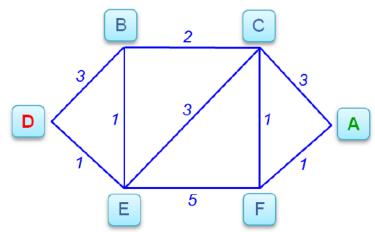
A - Le graphe étudié

A, B, C, D, E ef F sont des villes représentées par les sommets.

Le coefficient en italique associé à une route (représentée par une arête) représente la distance en km entre les deux villes adjacentes. La distance peut être remplacée par la durée ou le coût.



On veut trouver le chemin le plus court (en km) de D à A.

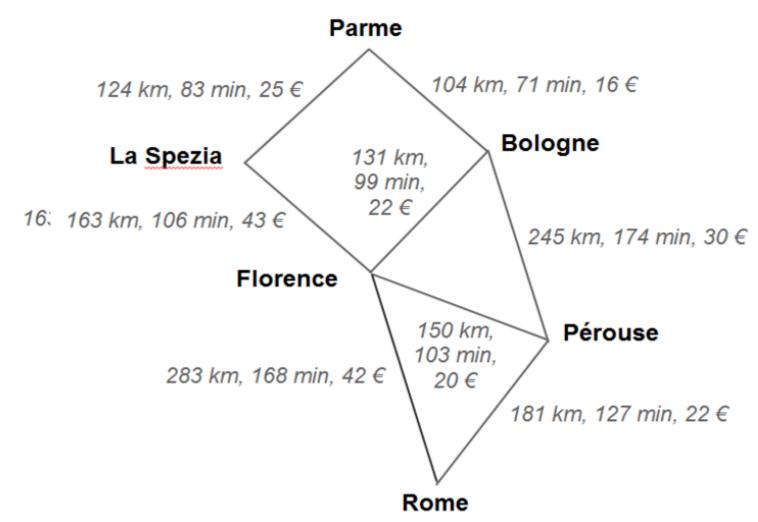


1 - Après la présentation du professeur, compléter, sur le fichier « Dijkstra1.ods », le déroulement de l'algorithme. L'algorithme s'arrête lorsque A devient la prochaine ville sélectionnée.

aller de D	2 - D'après le tableau que vous venez de remplir, quel est l'itinéraire le plus court pour à A ? Comment l'avez-vous obtenu ?
	3 - Écrire l'algorithme que vous avez utilisé.

II - Application à un cas concret

On cherche à se rendre de Parme à Rome. Le graphe utilisé est le suivant :



1 - En utilisant trois versions du fichier « Dijkstra2.odt », appliquer l'algorithme précédent pour déterminer successivement :

a - l'itinéraire le plus court,

 Itinéraire:
 Prix :

 Longueur:
 Durée :

 b - l'itinéraire le plus économique,

 Itinéraire:
 Prix :

 Longueur:
 Durée :

c - l'itinéraire le plus rapide

 Itinéraire:
 Prix :

 Longueur:
 Durée :

III - Programmation Python

On souhaite ici programmer cet algorithme.

A - Niveau A ***

A partir du fichier « Dijkstra A.py »:

- programmer l'algorithme de Dijkstra
- Reconstruire le chemin optimal
- L'afficher et afficher son coût.

B - Niveau B **

Dans le fichier « Dijkstra B.py », compléter le code afin d'obtenir l'algorithme de Dijkstra.