



GRAPHES PONDÉRÉS ET DIJKSTRA

TP 14

Graphes

- v1.0

Lycée La Martinière Monplaisir, 41 Rue Antoine Lumière, 69372 Lyon

1 Matrice de durée

On trouve, dans les atlas routiers par exemple, des matrices de durée de trajet permettant de connaître la durée de trajet typique entre 2 villes. Un exemple est donnée sur la FIGURE 1 provenant d'un vieil atlas routier Michelin.

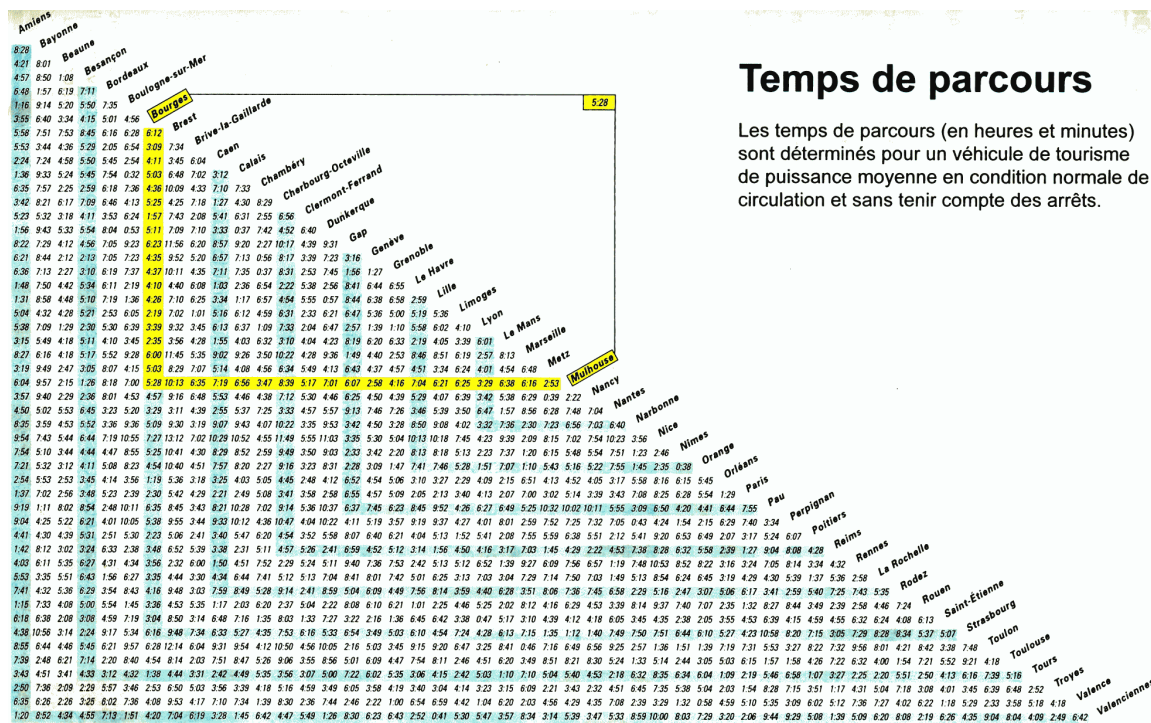


FIGURE 1 – Matrice de durée de trajet

Cela va nous permettre de définir un graphe pondéré complet avec les différentes villes comme nœuds et la pondération des arcs correspondant à la durée du trajet entre ces différents nœuds. Il est dit complet car chaque nœud possède une arête avec tous les autres nœuds.

Un algorithme de reconnaissance de texte (*Optical Character Recognition*) nous a permis d'obtenir le fichier `Mduree.csv` à partir de l'image présentée sur la FIGURE 1. Le script `GrapheTrajet_etudiants.py` permet de récupérer, à partir de ce fichier, la liste des 50 villes appelée `Villes` et une liste de listes `Mstr` contenant les chaînes de caractères correspondant à la durée du trajet sous la forme `hh:mm`.

Question 1 Écrire une fonction `hhmm2minutes(txt: str) -> int` qui prend en entrée une chaîne de caractères sous la forme `hh:mm` et qui renvoie la durée en minute.

Question 2 Vérifiez que `hhmm2minutes(Mstr[37][17])` renvoie bien l'entier 312. Il s'agit de la durée du trajet Reims-Grenoble.

La structure de données étant fondamentalement symétrique (il ne s'agit pas d'un graphe orienté), seule une moitié de la matrice est donnée.

Question 3 Créer un dictionnaire de dictionnaires `duree` tel que `duree[ville1][ville2]` nous donne la durée du trajet entre `ville1` et `ville2` en minutes. `ville1` et `ville2` sont des chaînes de caractères.

Question 4 Vérifiez que `duree["Reims"]["Grenoble"]` et `duree["Grenoble"]["Reims"]` renvoient tous les deux bien l'entier 312.

Si les durées correspondaient bien au temps de parcours le plus court possible, on aurait les durées qui correspondraient à une distance au sens mathématique du terme.



Définition Distance (mathématiques)

On appelle distance sur un ensemble E toute application d définie sur le produit $E^2 = E \times E$ et à valeurs dans l'ensemble \mathbb{R}^+ des réels positifs ou nuls,

$$d : E \times E \rightarrow \mathbb{R}^+$$

vérifiant les propriétés suivantes :

- Symétrie : $\forall (a, b) \in E^2, d(a, b) = d(b, a)$;
- Séparation $\forall (a, b) \in E^2, d(a, b) = 0 \Leftrightarrow a = b$;
- Inégalité triangulaire : $\forall (a, b, c) \in E^3, d(a, c) \leq d(a, b) + d(b, c)$.

Question 5 À partir de la représentation du graphe pondéré `duree`, vérifiez si on respecte l'inégalité triangulaire. Si l'inégalité triangulaire n'est pas respectée, vous afficherez sur la console tous les triplets de villes telles que `duree[ville1][ville3] > duree[ville1][ville2] + duree[ville2][ville3]`.

Question 6 Déterminer un triplet pour lequel l'écart est maximal.

Une autre représentation possible d'un graphe pondéré est d'avoir un dictionnaire d'adjacence `G` tel que `G[ville]` contienne une liste de couples (`successeur`, `poids`) au lieu de juste la liste des successeurs.

Question 7 Créer cette représentation sous forme de dictionnaire d'adjacence `Gduree`.

Question 6 Vérifiez que vous obtenez le chemin suivant pour aller de Reims à Grenoble : ['Reims', 'Troyes', 'Beaune', 'Lyon', 'Grenoble'].

Le script `GrapheDijkstra2_etudiants.py` permet de faire la même chose que précédemment, mais ajoute la possibilité de traiter une matrice `M` (matrice codée par un dictionnaire de dictionnaires) en plus du graphe `G`.

Question 7 Après avoir remplacé (copier/coller) votre implémentation de `popPrioritaire` dans le script, vérifiez le bon fonctionnement du script en testant `DureeDijkstraG` et `DureeDijkstraM`.

Question 8 Comparer les performances temporelles entre l'utilisation d'une matrice ou d'une structure d'adjacence. Est-ce que ces résultats sont surprenants ? (On utilisera la fonction `perf_counter_ns()` du module `time`).