

O

Lycee Charlemagne	MPSI2	Annee 2021/22
<h1 style="margin: 0;">Graphes</h1> <p style="margin: 0;">Mars/Avril</p>		

Le but de ce T.D. est d'étudier les graphes, et d'y définir quelques outils et algorithmes usuels.
Notre support sera le métro parisien, avec ses stations, reliées entre elles par des lignes.

Un graphe est un ensemble de points (les sommets)
d'arcs (vertex en anglais) reliant des sommets.

Dans les graphes orientés, les arcs le sont (*on peut aller de A à B, mais pas forcément de B à A*). Il peut donc y avoir zéro, un ou deux arcs entre deux sommets (*non reliés, sens unique, double sens*).

Dans les graphes non orientés, l'existence d'un arc entre A et B signifie qu'on peut aller de A à B comme de B à A.

Notre graphe du métro est un graphe orienté, pourquoi ? (*cherchez du côté des lignes « bis »*).

La convention est qu'il y a un arc de A à B si B est la station juste après A sur une ligne.¹
Dans ce cas, B est successeur de A et A est prédécesseur de B.

Quelle est la station qui a le plus de successeurs ?
Est ce la même que celle qui a le plus de prédécesseurs ?
Connaissez vous les stations sans successeur ni prédécesseur ?

Dans notre graphe, il y a par exemple deux lignes qui vont de RÉPUBLIQUE à STRASBOURG SAINT DENIS. On ne mettra qu'une arête de RÉPUBLIQUE à STRASBOURG - SAINT-DENIS.

C'est d'ailleurs une convention sur les graphes.

Lycee Charlemagne	MPSI2	Annee 2021/22
<h2 style="margin: 0;">Exemples</h2>		

Quelques situations où on utilise des graphes.

Les sommets sont des ordinateurs, reliés en réseau, et une arête est une connexion directe entre deux ordinateurs.

Pour l'ensemble MPSI2, les sommets sont les élèves, un arc entre A et B traduit « est dans le même trinôme que ».
Bon, c'est décevant.

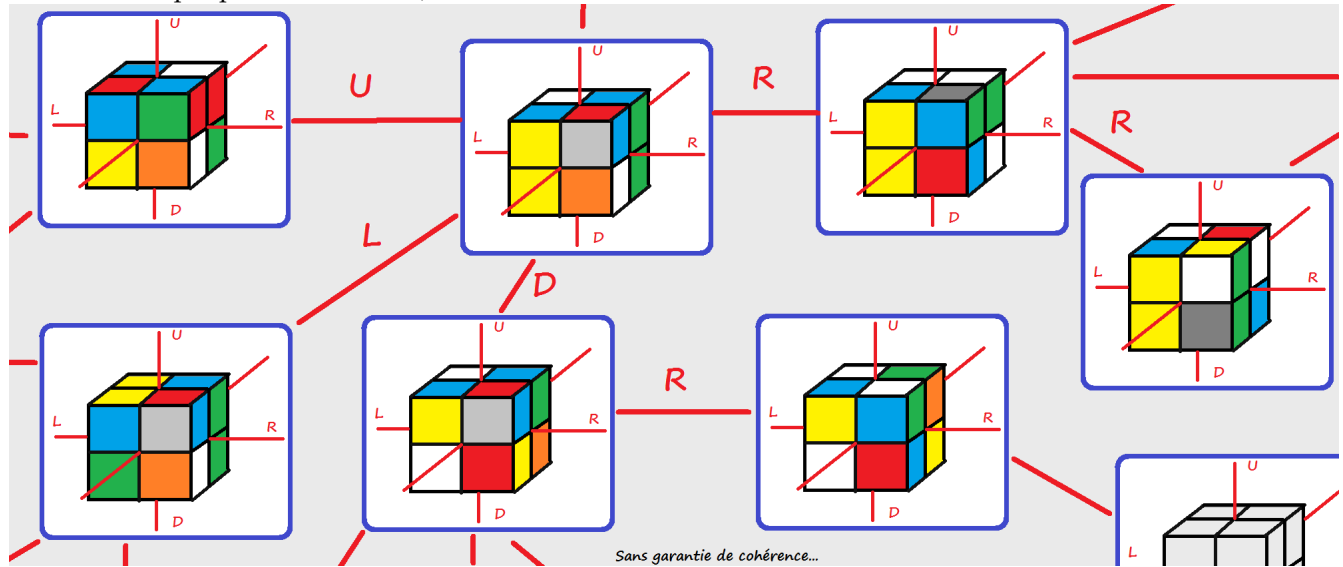
Pour l'ensemble MPSI2, un arc traduit « est sorti avec ». Est il orienté ? Existe-t-il des boucles ?²
Et si un arc traduit « a tenté de sortir avec », est il orienté. Qui a eu le plus de successeurs ? Qui a eu le plus de prédécesseurs ?

Prenons l'ensemble des états d'un Rubik's Cube de taille 2 sur 2. Les sommets sont les états possibles du cube. Un arc relie deux sommets si on peut passer d'un état à l'autre par un mouvement élémentaire (left, right, left', right',

1. il y a un arc de SAINT-PAUL à HÔTEL DE VILLE et un arc de SAINT-PAUL à BASTILLE ; un arc de BASTILLE à GARE DE LYON, mais pas d'arc de SAINT-PAUL à GARE DE LYON

2. une boucle est un arc qui va d'un sommet à lui même

front, front', up, up', down', down')³.

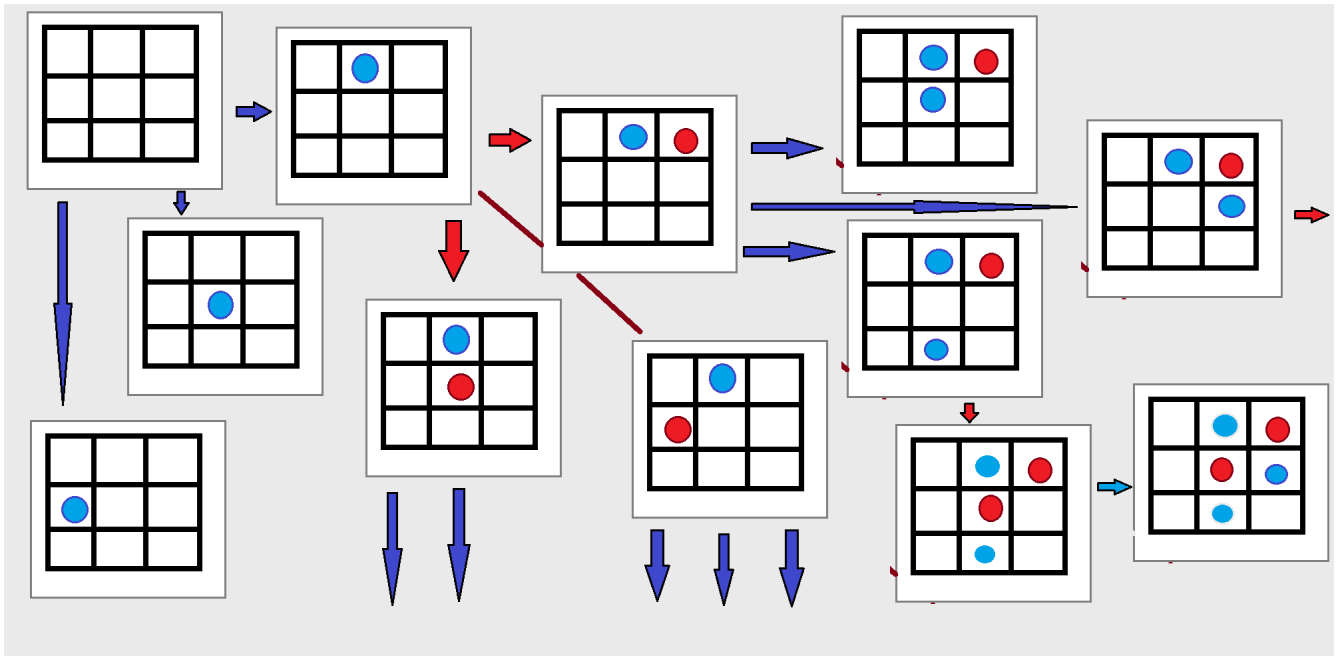


Combien de sommets (encadrement même grossier). Combien d'arcs ?

Résoudre le cube, c'est être capable de trouver un chemin⁴ qui va de l'état brouillé à l'état débrouillé. Est il possible qu'il n'y ait pas de chemin entre deux sommets (si vous disposez d'un tournevis).

Le jeu du « petit morpion » ou « tik-tak-toe » peut il être modélisé par un graphe (orienté) dont le schéma est ci-contre :

3. les mouvements sont des quarts de tour, et le prime est là pour indiquer qu'on tourne en sens inverse.
4. un chemin est une suite d'arcs où le successeur de l'arc d'indice i est le prédécesseur de l'arc d'indice i+1

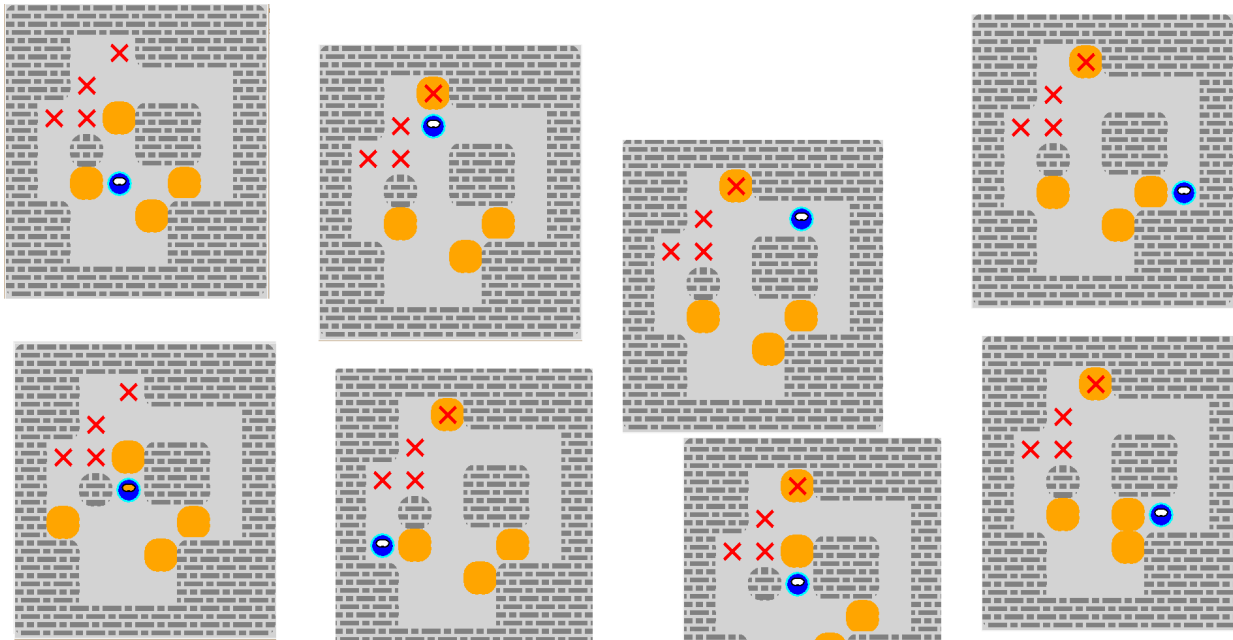


Si oui, combien de sommets ?

Quel est le sommet qui a le plus de successeurs ? Quelle est la longueur maximale d'un chemin ?

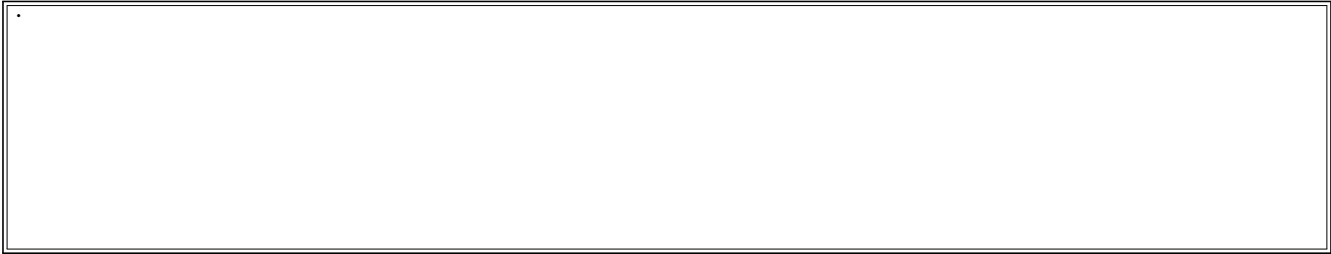
Qui sont les sommets sans successeurs ?

Il paraît que le sokoban (le petit bonhomme qui pousse des caisses case par case jusqu'à des emplacements prédéfinis) peut se représenter par un graphe.

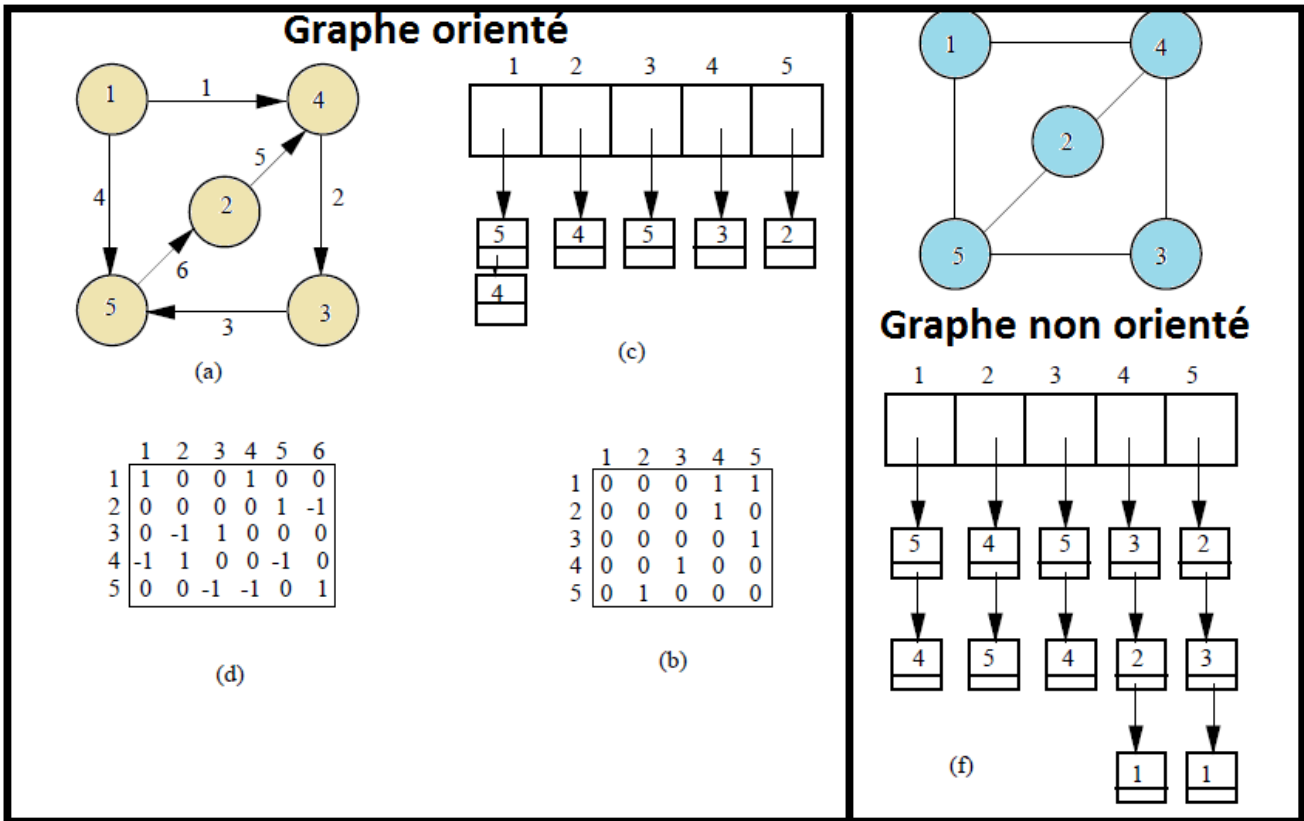


Reliez les sommets qui peuvent l'être. Indiquez ceux qui conduisent à un échec. Résolvez de tête ce Sokoban.

Si vous êtes intéressés, il y a un T.D. d'Éric Detrez, lycée Faidherbe à Lille. Expliquez cette phrase de son TD :
 « Dans la suite, nous allons nous intéresser uniquement aux mouvements des caisses, et donc considérer que le personnage peut se déplacer instantanément dans toute zone de cases vides contiguës. Nous allons pour cela cloner le personnage dans toute la zone de cases vides contiguës à laquelle il appartient. Compléter la fonction cases_libres_voisines, qui prend en arguments une carte et des entiers i, j correspondant aux coordonnées d'une case (qui n'est pas au bord de la carte), et renvoie la liste des coordonnées des cases vides voisines de $(i ; j)$. »

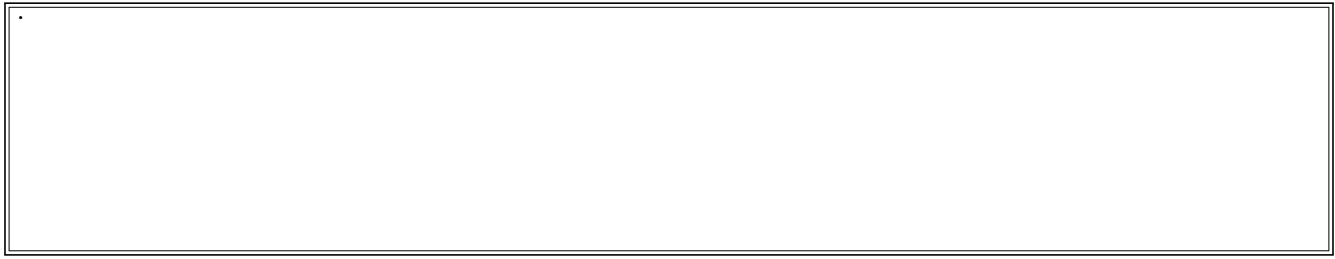


Non, même si on dirait des mots pour contrepèterie, c'est juste pour dire qu'un graphe se représente par une matrice. Ou des listes.



La matrice d'incidence C ayant autant de lignes que de sommets
 autant de colonnes que d'arcs
 des 0, des 1 et des -1
 Donnez la formule pour son terme général. Que pouvez vous dire de la somme des termes d'une d'une colonne ?
 de toute la matrice ?
 Comment interpréter une ligne dont tous les coefficients sont nuls ?
 A quoi reconnaît on alors un graphe plein.⁵
 Petit jeu : extrayez des matrices carrées, et calculez leur déterminant... Quelles sont les seules valeurs possibles ?

5. un graphe plein est un graphe où chaque sommet est relié à chaque sommet



La matrice d'adjacence M • carrée

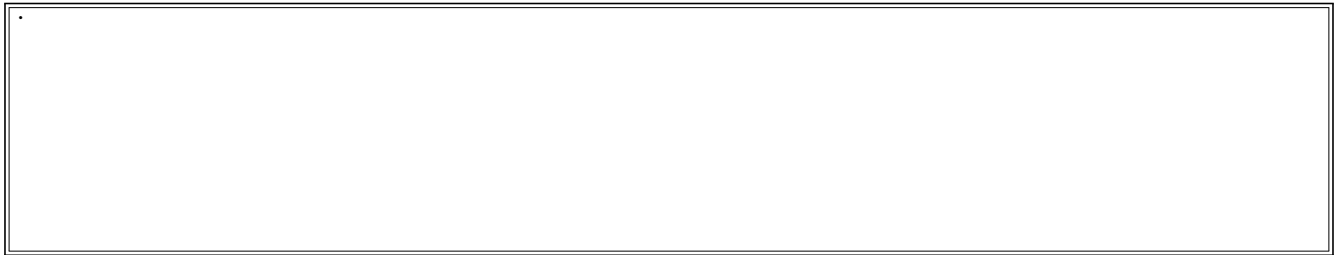
- ayant autant de lignes que de sommets
- $M_i^k = \begin{cases} True & \text{si il existe un arc de } i \text{ à } k \\ False & \text{sinon} \end{cases}$

Quel sera le format de la matrice d'adjacence du métro ?

Quelle sera sa trace ?

Est elle symétrique ?

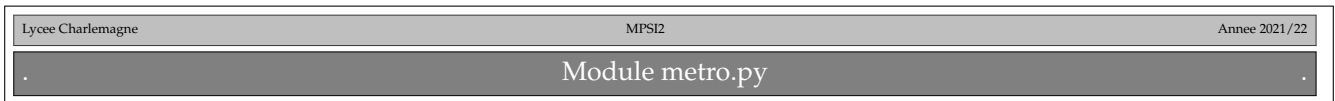
Que représente la somme des termes de cette matrice ?



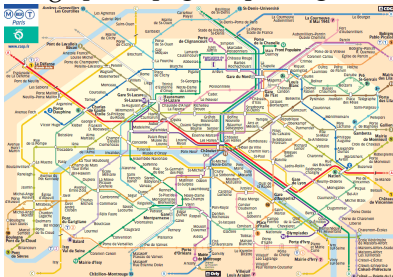
C'est elle qu'on va utiliser par la suite. Le module `metro.py` peut créer cette matrice et même la représenter.

Sinon, il peut être moins coûteux en termes de mémoire de se contenter de donner pour chaque sommet la liste de ses successeurs.

Là aussi, le module `metro.py` a justement pour chaque station la liste de ses successeurs.



Ce module a besoin d'un fichier d'un fichier texte : `RATP_voisins.txt`
image `plan.png` ou `plan.gif`



Colonel Fabien ; COLO ; 2 ; Paris 10e, Paris 18e ; 769 ; 219 ; ['jaur', 'bell']
 Commerce ; COMM ; 8 ; Paris 15e ; 244 ; 513 ; ['faur', 'lamo']
 Concorde ; CONC ; 1, 8, 12 ; Paris 8e ; 325 ; 359 ; ['tuil', 'clem', 'asse', 'made', 'inva', 'made']
 Convention ; CONV ; 12 ; Paris 15e ; 279 ; 597 ; ['pver', 'vaug']
 Coirentin Cariou ; CARI ; 7 ; Paris 19e ; 784 ; 130 ; ['crim', 'pvil']
 Coirentin Celton ; CELT ; 12 ; Issy-les-Moulineaux ; 223 ; 639 ; ['miss', 'pver']
 Corvisart ; CORV ; 6 ; Paris 13e ; 622 ; 581 ; ['pita', 'glac']
 Cour Saint-Émilien ; EMIL ; 14 ; Paris 12e ; 831 ; 570 ; ['berc', 'bnfm']

Quand vous l'exécutez, normalement, un plan de métro s'affiche⁶.

Vous pouvez l'utiliser pour cliquer sur une station qui vous intéresse.

Il vous donnera son nom et aussi son code.

En effet, chaque station a un numéro dans la liste (qui correspond à sa colonne et sa ligne dans la matrice), mais aussi un code de quatre lettres.

Et aussi un nom complet, à rallonge comme « SAINT-DENIS - AUBERVILLIERS - FRONT POPULAIRE ».

Un dictionnaire donne le code en fonction du numéro et un autre donne le numéro en fonction du code.

6. erreurs possibles : format du fichier image, essayez `plan.gif` ; encodage du fichier texte, ajoutez un `encoding='latin-1'`

Il y a Dico et Codi. Lequel va du sens code vers nombre, lequel va de nombre vers code.

0	nom	string
1	code	string longueur 4
2	adresse	string
3	lignes	liste d'entiers
4	abscisse sur le graphe	integer
5	ordonnée sur le graphe	integer
6	liste des successeurs	liste de codes

Les éléments de L sont des listes dont les éléments sont

Vérifiez avec L[0].

- 1- Trouvez la station dont le nom est le plus long.
- 2- Trouvez la station dont le nom est le plus court.
- 3- Trouvez la station la plus à l'ouest.
- 4- Sous quel numéro ai-je classé les lignes 7bis et 3bis ?

5- Quelle sera la réponse à

```
s = 0
for Stat in L :
...s += int('Saint' in Stat[0])
print(s)
```

Quelle est la station de métro de Paris intra-muros la plus loin de SAINT-PAUL⁷ ? ('Paris' in S[2] doit être un bon test pour ne garder que les stations dans Paris, non ?)

Que fait le script suivant :

```
def Scooby(n) :
...for S in L :
.....if n in S[3] :
.....A.append(S[0])
...A.sort()
...return(len(A), A)
```

J'ai oublié des stations ?

Il y a de nouvelles stations : LUCIE AUBRAC en prolongement de la ligne 4 à Bagneux (il y a aussi BARBARA, au moins le sais-tu ?).

Vous devez non seulement l'ajouter à la liste L. 1- Sous quelle forme ?

2- Mais que devez-vous aussi modifier dans L ?

7. POUR LA DISTANCE EUCLIDIENNE $\sqrt{x^2 + y^2}$

J'ai oublié aussi le prolongement de la ligne 14 au delà de SAINT-LAZARE.

Vous aimez les anciennes stations comme ARSENAL entre QUAI DE LA RAPÉE et BASTILLE.

3- Que devez vous ajouter ?

4- Que devez vous modifier ?

5- Sachant que RÉPUBLIQUE s'appelle 'repu' dans le dictionnaire, affichez la liste de ses successeurs.

6- Qui sont les stations n'ayant qu'un successeur ?

7- Que serait une station vérifiant $\text{len}(L[k][6]) = 0$?

8- Combien vaut $\text{sum}(\text{len}(L[k][6]) \text{ for } k \text{ in range}(n))$?

9- Une station est par erreur son propre successeur. Retrouvez la et corrigez le fichier `RATP_Voisins.txt`.

10- Affichez la liste des successeurs des successeurs de République.

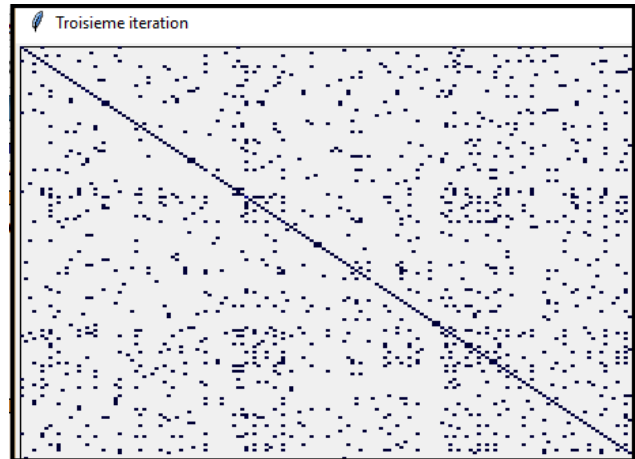
11- Est il normal d'y retrouver RÉPUBLIQUE elle même ?

12- Quelles sont les stations qui ne sont pas dans la liste des successeurs de leurs successeurs ?

A partir des listes de successeurs, créez la matrice d'incidence du graphe du métro (taille n sur n).

Demandez vous à PYTHON de l'afficher ?

Une procédure `AfficheMatrice()` est déjà saisie. Elle prend en entrée une matrice (et quelques options) et affiche la matrice « ve de loin », avec rien pour les 0, et un niveau de bleu plus ou moins poussé pour les autres termes. Sachant qu'il y a environ trois cent lignes, une case sera de taille 3 pixels sur deux pixels (les constantes 3 et 2 sont prises par défaut, mais vous pouvez en modifier la valeur suivant la résolution de votre écran). Vous pouvez aussi donner un titre à votre graphe.



- 1- Tiens, la matrice que j'affiche ci-contre a des termes sur la diagonale, est ce normal ?
- 2- Est-elle symétrique ?

Lycee Charlemagne

MPSI2

Année 2021 / 22

Algèbre linéaire.

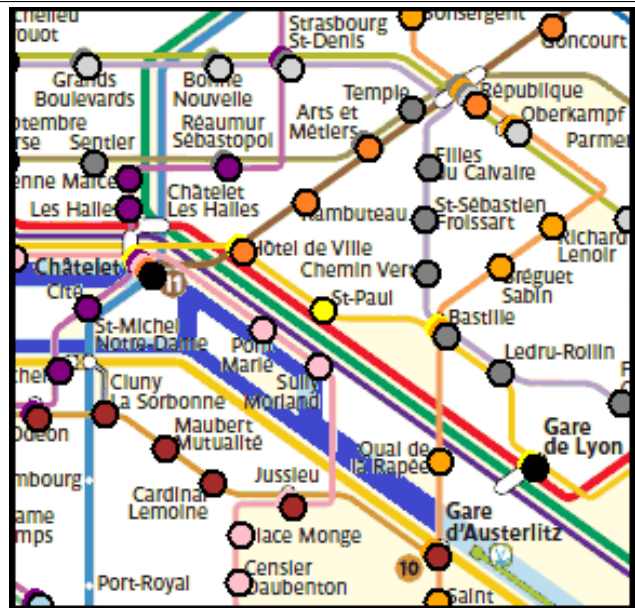
La matrice M est carrée. On peut alors l'élever au carré et calculer ses puissances.

	dans M	dans M^2	dans M^3
a_{paul}^{ramb}	0	1	0
a_{paul}^{paul}	0	2	0
a_{paul}^{repu}	0	0	0

Justifiez

Pour deux indices i et k fixés, combien de termes de la somme $\sum_k a_i^j \cdot b_j^k$ sont non nuls.

Pour quelle puissance de M le terme a_{paul}^{repu} vaudra autre chose que 0 ?



- 3- Complétez la procédure qui prend en entrée deux matrices et calcule leur produit :


```

def Prod(A, B) :
...LA, CA = len(A), len(A[0])
...LB, CB = len(B), len(B[0])
...assert CA == LB
...P = [[0 for k in range(CB)] for i in range(LA)]
...for i in range(LA) :
.....for k in range(CB) :
.....for j in range(CA) :
.....
...return(P)

```

4- Mettez en boucle votre procédure pour afficher la puissance dixième de la matrice d'incidence.

5- Pour quelle puissance le tableau sera-t-il totalement bleu ?

Quelles sont les stations les plus longues à relier de l'une à l'autre (à l'instinct).

Le temps d'exécution du produit matriciel et de l'affichage est long.

De plus, le produit matriciel revient sur ses pas.

Que penser effectivement du fait qu'un trajet SAINT-PAUL -> HÔTEL DE VILLE -> CHATELET -> HÔTEL DE VILLE -> SAINT-PAUL soit pris en compte.

Ou SAINT-PAUL -> HÔTEL DE VILLE -> CHÂTELET -> HÔTEL DE VILLE -> RAMBUTEAU.

Imaginons un individu qui part de SAINT-PAUL (*cliquez sur le plan pour connaître son code, et utilisez Dico pour connaître son index*).

Sur une page de gauche de son carnet, il écrit SAINT-PAUL (*ou plutôt son code*).

Sur la page de droite, il a la liste de toutes les stations, pour l'instant à distance infinie. Et il écrit que SAINT-PAUL est à distance 0.

Il regarde les stations auxquelles il peut accéder : HÔTEL DE VILLE et BASTILLE.

Sur la page de gauche de son carnet il ajoute ces deux stations et il efface SAINT-PAUL (*c'est bon, il a fini de la visiter*).

Et sur la page de droite, il marque que HÔTEL DE VILLE et BASTILLE ne sont plus à distance infinie mais à distance 1.

Et il regarde : depuis HÔTEL DE VILLE, où peut il aller ?

A SAINT-PAUL, à CHÂTELET et RAMBUTEAU.

Mais en lisant à droite, il sait qu'il connaît déjà SAINT-PAUL. Il n'y retournera pas.

En revanche, il veut bien ajouter CHÂTELET et RAMBUTEAU à la liste.

A distance 2.

Et c'est bon, il efface HÔTEL DE VILLE.

A ce stade :

BASTILLE, CHÂTELET et RAMBUTEAU.	toutes les stations à distance infinie, sauf SAINT-PAUL distance 0, HÔTEL DE VILLE et BASTILLE à distance 1, et CHÂTELET et RAMBUTEAU à distance 2.
----------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------

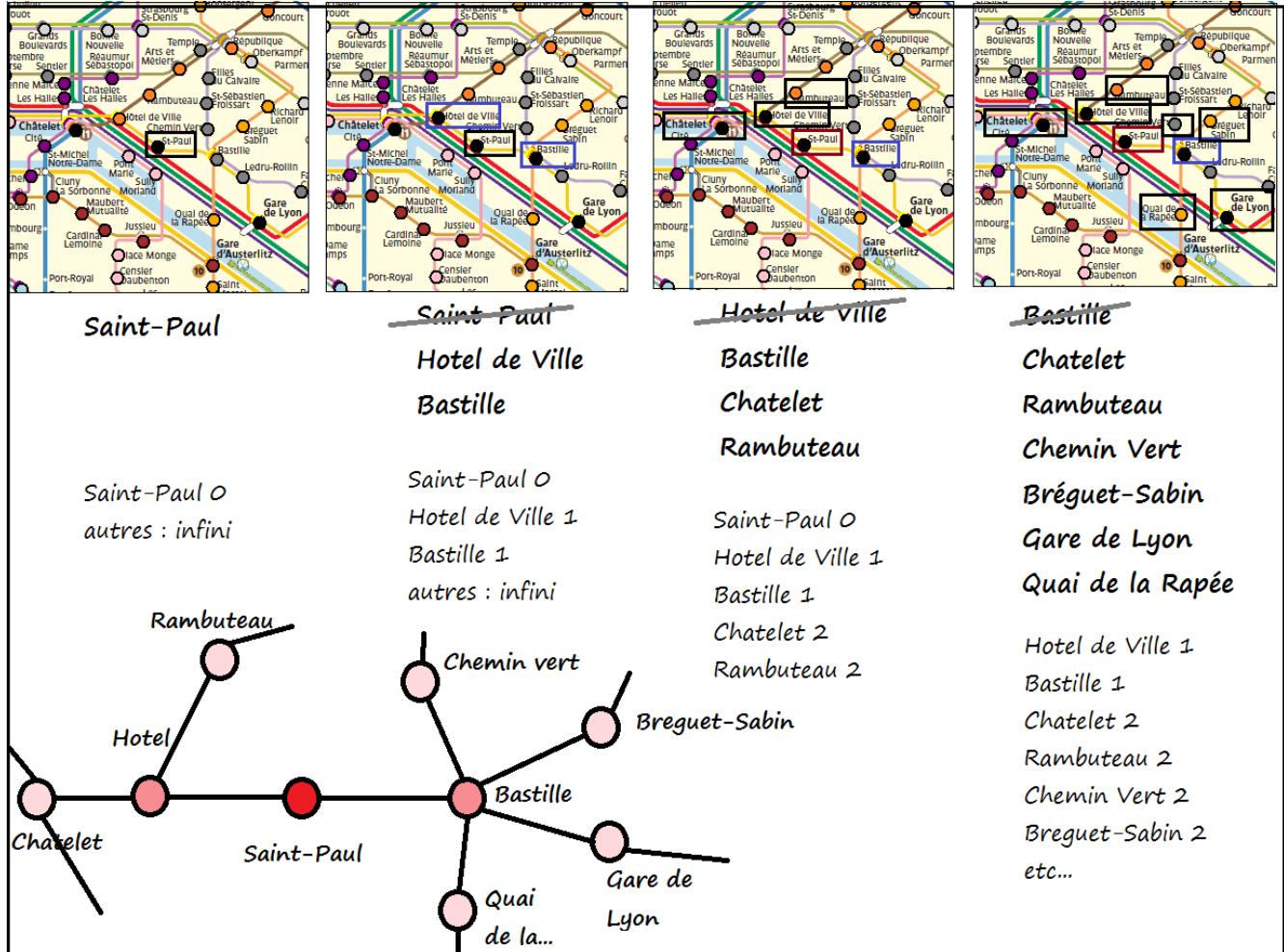
Et il regarde : depuis BASTILLE, où peut il aller.

A SAINT-PAUL (non ! déjà visitée), BRÉGUET, QUAI DE LA RAPÉE, GARE DE LYON, et que sais je encore.

Il efface BASTILLE de sa page de gauche, ajoute ces noms en fin de liste de la page de gauche.

Et il marque sur la page de droit qu'il sait comment y aller.

Et ainsi de suite.



Il y a donc une liste des stations à visiter⁸.

On l'agrandit par `append()`.

On enlève l'élément de tête par `pop(0)`.

Une autre liste contient les informations sur l'accessibilité des stations depuis SAINT-PAUL, avec l'infini pour les stations pas encore visitées, et la distance à SAINT-PAUL pour les autres.

Pour savoir si une station a été visitée ou non, on regarde sa distance à SAINT-PAUL...

On s'arrête quand un objectif initial a été atteint.

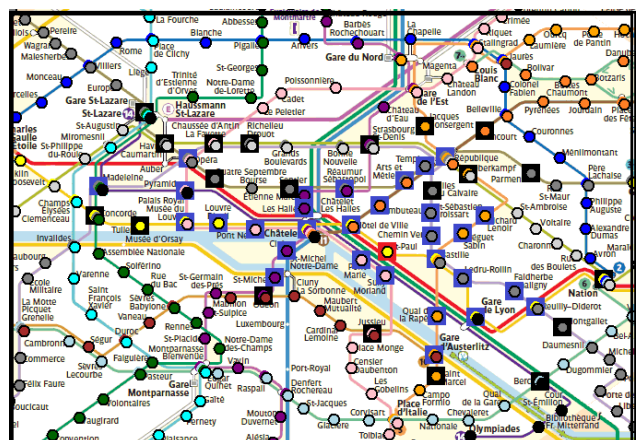
Par exemple votre domicile.

Ou plus de station à visiter.

Écrivez le script Python, en utilisant les `L[k] [6]` donnant les successeurs.

Une fonction offerte vous permet d'ailleurs de noircir sur le plan une station déjà visitée.

C'est `PointeStation()`, qui prend en entrée une station (*entier ou code, comme vous voulez*) et une couleur (*si vous n'en précisez pas, ce sera par défaut le noir*).



8. en fait une deque, le mot n'est pas de moi