NSI · TERMINALE / COURS

Les graphes

Astrolab Cours · astrolabcours.fr

01 Le problème

Un réseau social, des villes reliées par des routes, des pages web reliées par des liens : on veut savoir qui est relié à qui, trouver un chemin, détecter un cycle. Ni une liste ni un arbre ne représentent bien ces liens dans tous les sens.

01 L'idée : le plan de métro

Un plan de métro, ce sont des stations et des lignes qui les relient. Peu importe la distance réelle sur la carte : ce qui compte, c'est quelles stations sont reliées. Un graphe, c'est exactement ça.

01 Le vocabulaire

Un **graphe** est formé de **sommets** reliés par des **arêtes**.

- Dans un graphe **orienté**, les liens ont un sens : on parle d'**arcs** (A → B n'implique pas B → A).
- Deux sommets reliés sont **voisins** (ou adjacents).
- Le **degré** d'un sommet est son nombre de voisins.
- Un **chemin** est une suite de sommets reliés deux à deux ; un **cycle** est un chemin qui revient à son départ.
- Un graphe est **pondéré** quand chaque arête porte une valeur (distance, durée, coût).

01 Matrice d'adjacence

Les sommets sont numérotés de 0 à $n-1$. La case (i, j) vaut 1 s'il y a une arête de i vers j, 0 sinon.

$$M = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix}$$

Ici, 0 est relié à 1 et à 2. Pour un graphe non orienté, la matrice est **symétrique**.

Liste d'adjacence

Pour chaque sommet, on stocke la liste de ses voisins, en Python avec un dictionnaire :

```
g = {
    "A": ["B", "C"],
```

© Astrolab Cours · astrolabcours.fr/nsi/terminale/graphes/ · Tous droits réservés : usage personnel uniquement, republication et usage commercial interdits

```
    "B": ["A", "D"],
    "C": ["A", "D"],
    "D": ["B", "C"],
}
len(g["A"])   # degré de A : 2
```

Laquelle choisir ?

	Matrice	Liste d'adjacence
savoir si A et B sont reliés	immédiat	parcourir les voisins de A
mémoire	n^2 cases	proportionnelle au nombre d'arêtes

Pour un grand graphe avec peu de liens (un réseau social), la liste d'adjacence est bien plus économe.

Exemple guidé

Le graphe non orienté a quatre sommets et quatre arêtes : A–B, A–C, B–C, C–D. Représente-le des deux façons.

Matrice d'adjacence (ligne A, puis B, C, D) :

$$\begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

Liste d'adjacence :

```
g = {"A": ["B", "C"],
     "B": ["A", "C"],
     "C": ["A", "B", "D"],
     "D": ["C"]}
```

1. La matrice est **symétrique**, car le graphe n'est pas orienté.
2. Le **degré** de C vaut 3 : c'est la somme de sa ligne, ou `len(g["C"])`.
3. A, B, C forment un **cycle** : on part de A et on y revient sans repasser par une arête.
4. La matrice a 16 cases pour 4 arêtes. Pour un réseau de millions de sommets peu reliés, la liste est bien plus légère.

Retour au problème

Les sommets sont les personnes, les villes ou les pages, les arêtes sont les liens. Avec cette représentation, chercher un chemin ou un cycle devient un parcours du graphe.

Les pièges

- Graphe non orienté : écrire chaque arête dans les deux sens.
- Un arbre est un graphe particulier (connexe, sans cycle) : tous les graphes ne sont pas des arbres.
- Le degré d'un sommet ne dit rien de la longueur des arêtes.

© Astrolab Cours · astrolabcours.fr/nsi/terminale/graphes/ · Tous droits réservés : usage personnel uniquement, republication et usage commercial interdits

© Astrolab Cours · astrolabcours.fr/nsi/terminale/graphes/ · Tous droits réservés : usage personnel uniquement, republication et usage commercial interdits