

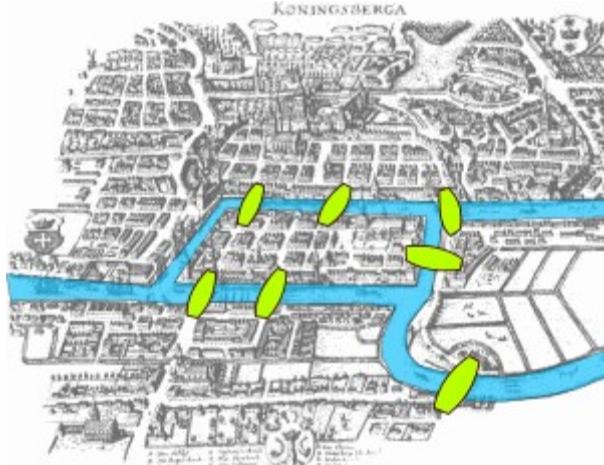
Type de document	Résumé de cours	Classe	Tle	Durée	2h	Date	15/05/2023
Thème et contenu(s)	Structures de données : graphes						
Capacités attendues	Connaître le vocabulaire de base et les différents types de graphes. Savoir les implémenter						
Prérequis	Listes à 2 dimensions, dictionnaires, objets						
Description	Cours d'introduction sur les graphes						

I) Les graphes : introduction

Les graphes sont des objets mathématiques simples à représenter, mais parfois complexes à étudier. De par leur construction, ils permettent de **modéliser des relations**.

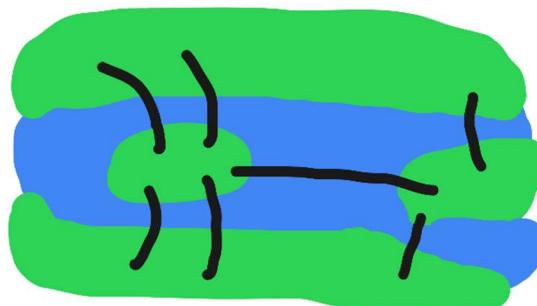
Leur étude commence au XVIII^e siècle quand Leonard Euler décrit le problème des « 7 ponts de Königsberg ». Le problème se pose de la manière suivante : existe-t-il un chemin permettant aux habitants, à partir d'un point de départ donné, de se promener à travers la ville en empruntant une et une seule fois chaque pont, puis de revenir à leur point de départ.

Voici un plan de la ville de Königsberg (enclave Russe de Kaliningrad) :

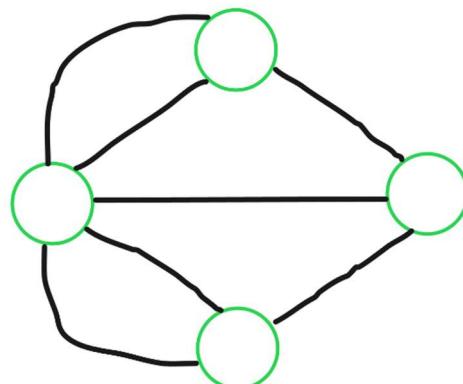


Représentation de Königsberg (source : Wikipédia)

Voici un magnifique schéma qui se concentre sur les rives et les ponts :



D'un tel schéma on peut extraire un **graphe** dans lequel les ponts sont représentés par des **arêtes** et les rives par des **sommets** :



On représente les sommets par des cercles et les arêtes par des traits.

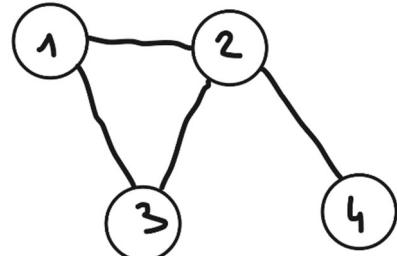
À ce stade, il faut comprendre qu'on peut représenter graphiquement un graphe de plusieurs manières différentes : tant que ces représentations comportent les mêmes sommets et les mêmes arêtes, ces graphes sont strictement identiques, bien que d'apparences différentes.

II) Définitions

a) Graphes non orientés

Un **graphe non-orienté** est un ensemble fini d'éléments $G=\{S, A\}$ où :

- S est un ensemble de sommets
- A est un ensemble d'arêtes (paires non ordonnées d'éléments de S)



On décrit la structure d'un graphe en **énumérant ses sommets et ses arêtes** en se basant sur la définition. Exemple avec le graphe suivant :

- $S = \{1, 2, 3, 4\}$
- $A = \{(1,2); (1,3); (2,3); (2,4)\}$

Dans le cas d'un graphe non orienté, une arête entre deux sommets peut se noter « dans les deux sens ». Ainsi, $(1,2)$ équivaut à $(2,1)$. En revanche, dans un graphe orienté, le sens a son importance.

b) Graphes orientés

III) Représentations

Tout graphe (orienté ou non orienté) peut être décrit avec une **représentation matricielle**. Vous devez être capable de donner la matrice d'un graphe à partir de sa représentation sommets/arcs(arêtes), et inversement.

Par exemple, la représentation matricielle du graphe précédent est :

	1	2	3	4
1	0	1	1	0
2	1	0	1	1
3	1	1	0	0
4	0	1	0	0

On met un 1 si les deux sommets sont reliés, un 0 sinon. Il y a des « doublons » dans le cas d'un graphe non orienté car le lien entre deux sommets est réflexif. On note aussi une **symétrie par rapport à la diagonale de la matrice**.

En Python, on peut représenter la **matrice d'adjacence** avec une **liste d'adjacence**. Soit on utilise une liste à deux dimensions et on énumère spécifiquement tous les liens possibles, soit on se sert d'un dictionnaire :

```
graphe = {}  
graphe[1]=[2, 3]
```

```
graphe[2]=[1,3,4]
graphe[3]=[1,2]
graphe[4]=[2]
```

Une notation plus concise et équivalente serait :

```
graphe={1: [2, 3], 2: [1, 3, 4], 3: [1, 2], 4: [2]}
```

IV) Vocabulaire

Ordre d'un graphe : nombre de sommets de ce graphe

Graphe complet : tous ses sommets sont adjacents (chaque couple de sommets est lié par une arête). S'il est d'ordre n , on a $n(n-1)/2$ arêtes

Degré d'un sommet :

- nombre d'arêtes liées à un sommet (graphe non orienté)
- somme des arcs entrants et sortants (graphe orienté)

cycle (graphe non orienté) ou chemin (graphe orienté) : suite consécutives d'arêtes (resp. d'arcs) avec deux sommets identiques

Graphe connexe : graphe dans lequel tout sommet peut être relié à un autre par une arête (ou une suite d'arêtes)

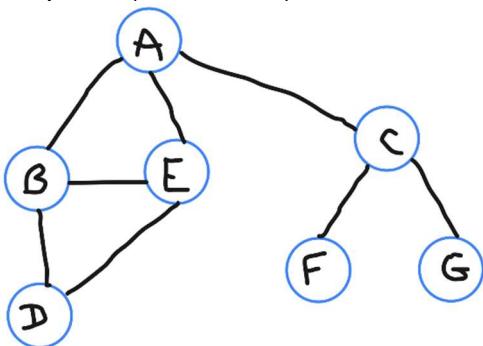
V) Parcours

c) En largeur (BFS)

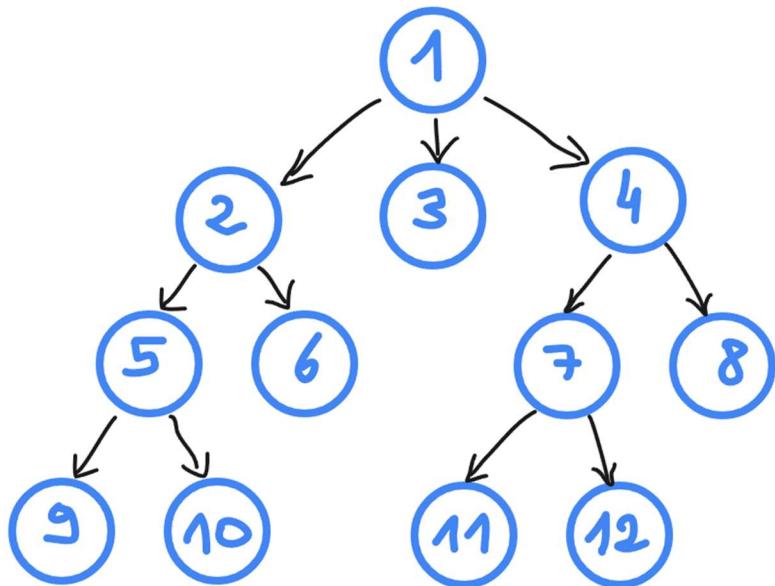
L'algorithme fonctionne un peu comme celui vu pour les ABR. Mais comme un graphe, à la différence d'un arbre, peut être cyclique, il faut faire attention de ne pas tomber dans une boucle infinie en visitant sans cesse les sommets déjà visités. On va donc, en plus d'une file comme pour les ABR, utiliser une liste afin de conserver/mémoriser les sommets déjà visités.

Graphe 1 : exemple de cours

Graphe 2 (non orienté) :



Graphe 3 (orienté, dans le style d'un arbre) :



Voir exercices pour l'implémentation