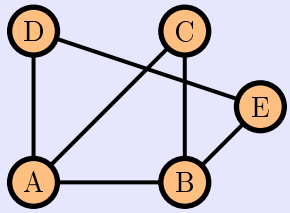
Graphes
11

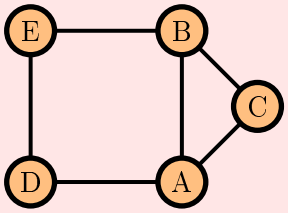
Définition

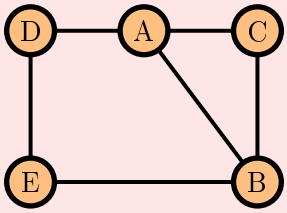


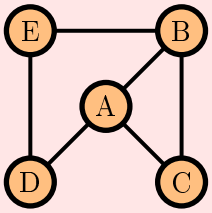
Un graphe est composé de **sommets** (*vertex*) liés par des **arêtes** (*edges*). Ainsi, on pourra noter $G(\mathcal{S}, \mathcal{A})$ pour désigner le graphe G qui possède un ensemble $\mathcal{S}$ de sommets et un ensemble $\mathcal{A}$ d'arêtes (*en anglais*, $G(\mathcal{V}, \mathcal{E})$).

Ici, $\mathcal{S} = \{A, B, C, D, E\}$ et $\mathcal{A} = \{(A, B), (A, C), (A, D), (B, C), (B, E), (E, D)\}$.

Dans un graphe, seuls comptent les sommets et leurs connexions.
Par exemple Ces graphes sont équivalents au précédent :



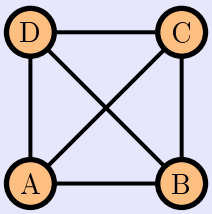




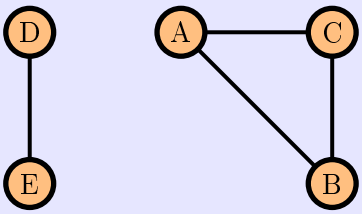
Définitions

- Le **degré** d'un sommet est le nombre d'arête ayant ce sommet pour extrémité.
  Le degré de A est 3.
- L'**ordre** d'un graphe est le nombre de sommets. *Le graphe est d'ordre 5.*
- Deux sommets sont **adjacents** s'ils liés par une arête.
  A et B sont adjacents mais pas A et E.
- Un chemin ou chaîne, est une suite de sommets adjacents décrivant un chemin dans le graphe.
  E-B-C-A-B est un chemin mais pas E-B-D-E car il n'y a pas d'arête (B,D).
- Une chaîne **fermée** est un chemin dont le premier point est égal au dernier. Une chaîne fermée où aucun sommet n'est répété (sauf le point de départ) est un **cycle**.
  E-B-A-D est une chaîne fermée et un cycle.
  E-B-A-C-B-A-D-E est toujours une chaîne fermée mais pas un cycle – peut dépendre des définitions.
- Un graphe **complet** est ou chaque sommet est adjacent avec tous les autres.
- Un graphe **connexe** est un graphe dans lequel on peut trouver un chemin reliant toute paire de sommets.

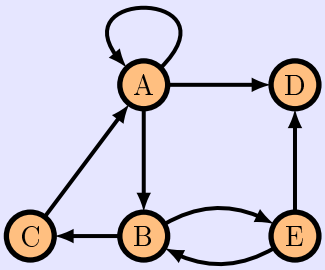
complet



non connexe



Graphe orienté



Si les arêtes sont dotées d'une orientation, on dit que le graphe est **orienté**.

Ci-contre, le passage C $\to$ A est possible mais pas A $\to$ C.

Le parcours en double sens B $\leftrightarrows$ E nécessite deux arêtes.

A $\to$ B $\to$ C $\to$ A est un cycle mais pas A $\to$ D $\leftarrow$ E $\to$ B $\leftarrow$ A.

On peut envisager le cas d'un sommet pointant sur lui-même.

On peut arriver à D mais pas en partir.

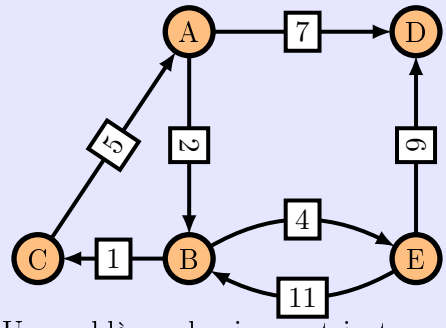
Matrice d'adjacence

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

La matrice est une grille représentant l'existence de connexion d'un sommet à l'autre. Nous avons n sommets. La matrice sera une grille de $n \times n$. La chiffre à la ligne i et la colonne j est 1 si l'arête sommet $i \to$ sommet j existe.

Ainsi le $\boxed{1}$ représente la l'arête E $\to$ B.

Graphe pondéré



Les arêtes sont pondérées par un nombre, un poids.

Cet exemple est orienté mais on pourrait pondérer un graphe non-orienté.

Par exemple, si les sommets représentent des lieux dans une ville, les arêtes des rues, les pondération pourraient représenter un temps de trajet. Il y a des sens unique et le parcours en sens contraire peut-être plus long.

Un problème classique est justement de chercher le chemin le plus court d'un point à un autre.

Interface

Si on souhaite implémenter des graphes dans un langage il faudra d'abord faire des choix : *orienté* / *non-orienté* et *pondéré* / *non pondéré*. On peut concevoir une implémentation polyvalente capable de gérer tous les cas mais ce n'est pas forcément le mieux.

Ensuite l'implémentation devrait permettre les méthodes suivantes :

- création d'un graphe vide,
- ajout d'un sommet avec son contenu,
- ajout d'une arrête entre deux sommets, pondéré / orienté suivant le cas,
- calcul du degré d'un sommet,
- calcul de l'ordre du graphe,
- test si une arête existe entre deux sommets,
- calcul de la matrice d'adjacence
- ensemble des successeurs d'un sommet,
- ensemble des antécédents d'un sommet