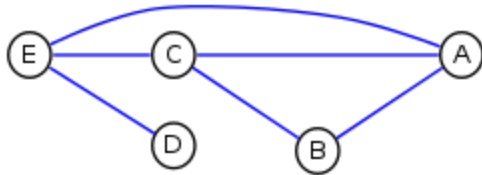


Les graphes

Laurent Signac

Un graphe est formé de noeuds et d'arêtes :



Il y a plusieurs familles de graphes :

- graphes orientés ou non
- graphes valués ou non
- graphes avec ou sans boucles
- multigraphes
- ...

Un des projets utilise des **graphes non orientés, non valués**.

Quelques définitions

un graphe non orienté non valué G est un couple $G = (X, E)$

- X est l'ensemble non vide des nœuds.
- E est l'ensemble des arêtes, paires (non ordonnées) d'éléments de X .

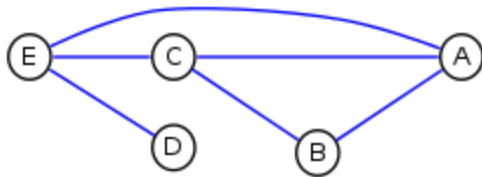
Vocabulaire sur les graphes :

- voisins d'un nœud : $x, y \in X$ sont voisins (ou adjacents) si $(x, y) \in E$
- sous-graphe d'un graphe : graphe (X', E') tel que $X' \subset X$ et $E' \subset E$
- ordre d'un graphe : nombre de nœud
- degré d'un nœud : nombre d'arêtes issues d'un nœud

Représenter un graphe

(non orienté, non valué)

Matrice d'adjacence 1



A	0	1	1	0	1
B	1	0	1	0	0
C	1	1	0	0	1
D	0	0	0	0	1
E	1	0	1	1	0

matrice carrée

symétrique

binaire

Il y a un 1 en (i, j) si i et j sont reliés et 0 sinon.

Adaptable à des graphes orientés (la matrice n'est plus symétrique)
et valués (la matrice n'est plus binaire).

Matrice d'adjacence 2

A	0	1	1	0	1
B	1	0	1	0	0
C	1	1	0	0	1
D	0	0	0	0	1
E	1	0	1	1	0

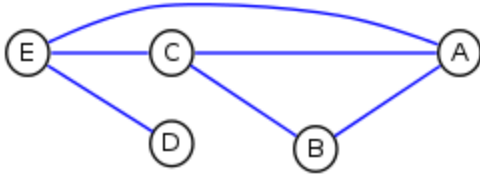
Inconvénients :

- la matrice est très creuse
- énumération des voisins pas très efficace

Avantages :

- on teste une arête en $O(1)$

Listes d'adjacence



```
A : B, C, E
B : A, C
C : A, B, E
D : E
E : A, C, D
```

On associe à chaque sommet la liste des ses voisins.

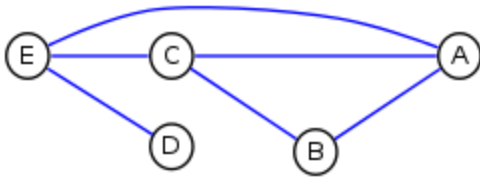
Inconvénients :

- on ne teste plus une arête en $O(1)$

Avantages :

- économie en place
- efficacité de l'énumération des voisin

Matrice d'incidence 1

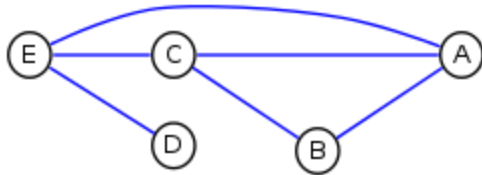


A	1	1	1	0	0	0
B	0	0	1	1	0	0
C	0	1	0	1	1	0
D	0	0	0	0	0	1
E	1	0	0	0	1	1

Graphe de n nœuds et p arêtes $\Rightarrow$ matrice de n lignes et p colonnes.

$M(i, j) = 1$ ssi l'arête j a le sommet i pour extrémité.

Matrice d'incidence 2



- mémoire en $O(np)$
- énumération des voisins $O(p)$
- on ne teste plus une arête en $O(1)$

Et nous ?

Dans un langage de haut niveau, il y a un compromis à trouver entre efficacité et simplicité d'utilisation.

Les graphes que nous manipulerons ne sont pas gigantesques.

⇒ **On met l'accent sur la simplicité** (si on a le reste, tant mieux).

Bonne structure de données

⇒ 75 % du boulot est fait

Exemple avec Python

La liste d'adjacence devient un dictionnaire d'ensembles !

```
g = {'A': {'B', 'C'},  
     'B': {'A', 'C'},  
     'C': {'B', 'E', 'A'},  
     'E': {'C'}}  
}
```

Défaut : Il faut veiller à la consistance de la représentation (chaque arête est en double).

Exemple avec Python : opérations

- ordre du graphe : `len(g)`
- degré du nœud n : `len(g[n])`
- énumérer les voisins de n : `for v in g[n]`
- présence d'une arête n_1, n_2 ? : `if n1 in g[n2]`

Les graphes dans le projet

- ➔ Représentations par dictionnaire d'ensembles
- ➔ À chaque nœud, on associe un couple de coordonnées
- ➔ À chaque nœud, on associe une valeur (qui ne peut pas être l'étiquette)
- ➔ Création d'une classe graphe (evt classe nœud)

Les graphes en général

- ont des applications pratiques (GPS etc ...)
- ne sont pas très complexes à mettre en œuvre avec un langage expressif ou une bibliothèque adaptée
- permettent de faire de l'algorithmique intéressante
- sont utiles pour les jeux

Problèmes sur les graphes

- recherche d'un chemin (existence, longueur minimale, longueur maximale dans un graphe sans boucle)
- recherche d'un parcours eulérien (on passe une fois et une seule par chaque arête)
- recherche d'un parcours hamiltonien (on passe une fois et une seule par chaque nœud)
- recherche du flot maximal dans un graphe orienté
- problèmes de coloration
- Les différentes configuration d'un jeu sont souvent représentables par les nœuds d'un graphe.

Bibliographie

- *Collectif* (2015). **Les graphes**. Bibliothèque Tangente HS 54, Éditions Pôle.
- *Cogis, O. et Robert, C.* (2003). **Théorie des graphes : problèmes, théorèmes, algorithmes**. Vuibert
- *Müller, D.* (2011). **Introduction à la théorie des graphes**, Cahier 6 de la CRM, <http://www.apprendre-en-ligne.net/graphes/>
- *Gondran M., Minoux M.* (2009), **Graphes et algorithmes**, 4e éd., Lavoisier