Ch.2 Représentations d'un graphe en machine

1. Par liste d'arêtes ou d'arcs
2. Par matrice d'adjacence
3. Par liste d'adjacence
4. Représentation des chemins et des arborescences

Représentation par liste d'arêtes

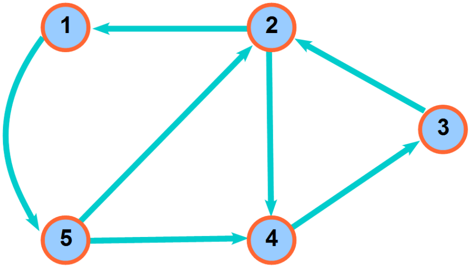
Définition

La représentation d'un graphe G par sa liste d'arête consiste à représenter G par son nombre de sommets et la liste des arêtes ou des arcs de G.

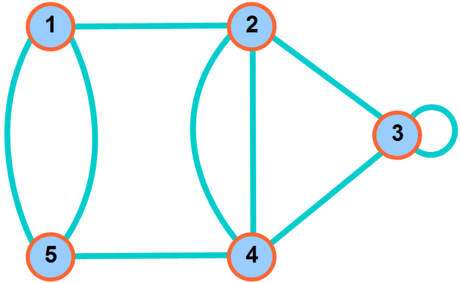
Remarque : La représentation de G par sa liste d'arête est "naturelle" d'après la définition d'un graphe, mais nous verrons qu'elle est peu efficace en pratique. Nous lui préférerons les deux représentations suivantes. En algorithmique elle est en général seulement utilisée pour la saisie du graphe et immédiatement transformée en représentation par matrice ou liste d'adjacence.

Le graphe orienté G_1 est représenté par la liste d'arcs

$$[[1,5],[2,1],[2,4],[3,2],[4,3],[5,2],[5,4]].$$



Le multigraphe non orienté G_2 est représenté par la liste d'arêtes $[[1,2],[1,5],[1,5],[2,4],[2,4],[2,3],[3,3],[3,4],[4,5]]$.



Représentation matricielle

Définition

On appelle matrice d'adjacence de G la matrice $A = (a_{i,j})$ *telle que pour tout couple* (i,j) *de* X^2 :

- *Dans le cas d'un graphe non orienté :* $a_{i,j}$ *est égal à 1 si* $\{i,j\} \in E$ *et 0 sinon. On a donc toujours* $a_{i,j} = a_{j,i}$.
- *Dans le cas d'un graphe orienté :* $a_{i,j}$ *est égal à 1 si* $(i,j) \in E$ *et 0 sinon. La matrice n'est donc plus nécessairement symétrique.*
- *Dans le cas d'un multigraphe :* $a_{i,j}$ *est le nombre d'arêtes ou d'arcs entre les sommets i et j (et les coefficients diagonaux comptent alors les boucles.)*

Représentation matricielle d'un graphe non orienté

Proposition

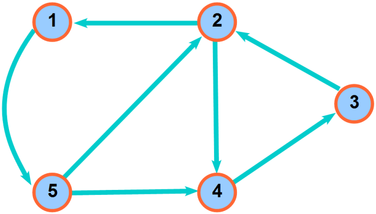
1. *La matrice d'adjacence d'un graphe non orienté est toujours symétrique.*
2. *La somme des coefficients de A est égale à deux fois le nombre d'arêtes m.*
3. *La somme des éléments de la ligne i ou de la colonne i est égale à $d(i)$.*

Représentation matricielle d'un graphe orienté

Proposition

1. *La somme des coefficients de A est égale au nombre d'arcs m.*
2. *La somme des éléments de la ligne i est égale à $d^{+}(i)$.*
3. *La somme des éléments de la colonne i est égale à $d^{-}(i)$.*

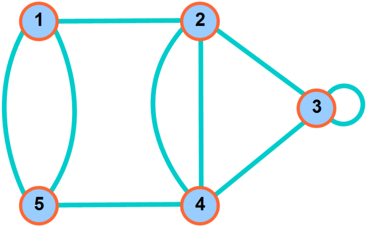
La matrice d'adjacence du graphe non orienté G_1



$$M_1 = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \end{pmatrix}.$$

On voit qu'il n'y a pas de boucle car les coefficients diagonaux sont tous non nuls.

La matrice d'adjacence du multigraphe non orienté G_2



$$M_2 = \begin{pmatrix} 0 & 1 & 0 & 0 & 2 \\ 1 & 0 & 1 & 2 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 2 & 1 & 0 & 1 \\ 2 & 0 & 0 & 1 & 0 \end{pmatrix}.$$

On voit qu'il y a une boucle et des arêtes multiples.

Cette représentation permet d'utiliser les opérations matricielles et les propriétés du calcul matriciel, elle est donc très intéressante théoriquement :

Proposition

On considère la matrice A^k *obtenue en multipliant* $k-1$ *fois la matrice* A *par elle-même. Alors le coefficient d'indice* (x, y) *de la matrice* A^k *est le nombre de chemins de longueur* k *entre les sommets* x *et* y.

Algorithmiquement, cette représentation est pratique par exemple pour tester l'existence d'un arc entre deux sommets (Si A[i,j] alors ...), pour ajouter ou retirer un arc (mettre A[i,j] à 0 ou 1), ce qui se fait en temps constant $O(1)$. Il est également facile de parcourir ou de traiter tous les sommets adjacents d'un sommets (en $O(n)$.)

```
{traiter les voisins de i, i.e. parcourir la ligne i}
Pour j de 1 à n faire
   si A[i,j] alors traiter j;
```

Exemple : Calculer le degré de tous les sommets .

```
Pour i de 1 à n faire {pour chaque sommet}
Début
D[x]:=0;
Pour j de 1 à n faire
    si A[i,j] alors D[x]:=D[x]+1;
  Fin;
```

On voit sur cet exemple l'inconvénient de cette représentation, les algorithmes de parcours sont en $O(n)$ quel que soit le nombre de prédécesseurs ou de successeurs de x. Une consultation de tous les arcs nécessite de visiter toutes la matrice donc un algorithme en $O(n^2)$. L'espace mémoire nécessaire est aussi en $O(n^2)$ quel que soit le nombre d'arêtes. La représentation suivantes sera plus économe.

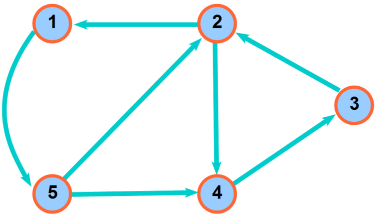
Représentation par les listes d'adjacence

Définition

La représentation de G par listes d'adjacence consiste à représenter G par un vecteur L indexé par X tel que pour tout sommet x, $L[x]$ soit la liste de ses successeurs dans G.

Remarque : Cette définition s'adapte au cas des graphes orientés ou non. Dans le cas d'un graphe non orienté, il faut bien vérifier que si y appartient à $L[x]$, x doit appartenir à $L[y]$.

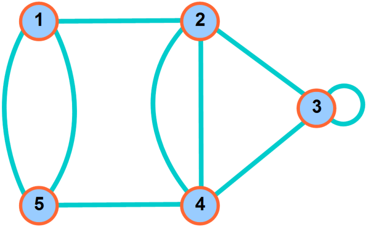
Exemple



Le graphe orienté G_1 est représenté par les liste d'adjacence

$$[[5],[1,4],[2],[3],[2,4]].$$

Exemple



Le multigraphe non orienté G_2 est représenté par les liste d'adjacence

$$[[2,5,5],[1,3,4,4],[2,3,4],[2,2,3,5],[1,1,4]].$$

Proposition

Pour un graphe à n sommets et m arcs, l'espace mémoire utilisé est un $O(n+m) = O(max(n,m))$.

L'algorithme de parcours des voisins est cette fois en $O(d(x))$ ou $O(d^+(x))$.

```
Pour tout y dans L[x] faire
   traiter y
```

Exemple : Calcul du degré :

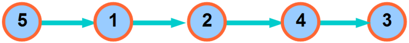
```
Pour i de 1 à n faire
  Début
    D[x]:=0;
    Pour y dans L[x] faire
       D[x]:=D[x]+1;
  Fin;
```

Ces performances sont telles que nous n'utiliserons en TP que la représentation par listes d'adjacences.

Représentation des chemins

- Le chemin $(x_0, x_1, ...x_n)$ peut être représenté par la liste $[x_0, x_1, ..., x_n]$ ou par un tableau des pères donnant pour chaque sommet du chemin le nom de son prédécesseur dans le chemin. Plus précisément :
  - Si x n'appartient pas au chemin alors $Pere[x] = 0$
  - $Pere[x_0] = x_0$ (l'origine est son propre père)
  - Pour les autres sommets du chemin, $Pere[x_{k+1}] = x_k$.

Le chemin



peut être représenté par la liste des sommets :

$$[5, 1, 2, 4, 3]$$

on son tableau des pères :

$$[5, 1, 4, 2, 5]$$

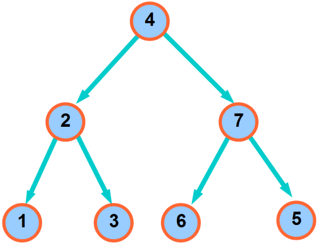
(Le père de 1 est 5, le père de 2 est 1, les père de 3 est 4, 5 est son propre père etc.)

Représentation des arborescences.

La propriété d'unicité du père dans une arborescence permet également de représenter une arborescence à l'aide d'un tableau des pères tel que :

- $Pere[r] = x_0$ (la racine est son propre père)
- Pour les autres sommets x de l'arborescence, $Pere[x]$ est le numéro de son père dans l'arborescence.

L’arborescence



peut être représenté par son tableau des pères :

$$[2, 4, 2, 4, 7, 7, 4]$$

(Le père de 1 est 2, le père de 2 est 4, les père de 3 est 2, 4 est son propre père etc.)