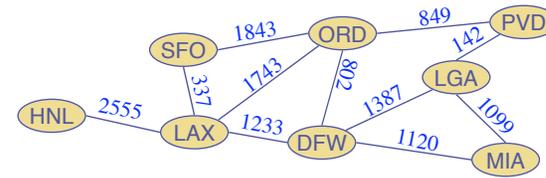


Graphes et structures de données pour les graphes

Graphes

- Un graphe est une paire (S,A) , où
 - S est un ensemble de noeuds appelés **sommets**
 - A est un multi-ensemble de paires de sommets appelées **arêtes**
 - On peut voir les sommets et les arêtes comme des positions gardant des objets en mémoire
- Exemple:
 - Chaque sommet représente un aéroport et garde en mémoire le code de 3 lettres représentant cet aéroport
 - Chaque arête représente une route aérienne entre deux villes et garde en mémoire le longueur de cette route



Types d'arêtes

● Arête orientée

- Une paire ordonnée de sommets (u,v)
- Le premier sommet u est appelé l'**origine**
- Le deuxième sommet v est appelé la **destination**
- Exemple: Un vol d'avion



● Arête non-orientée

- Une paire non-ordonnée de sommets (u,v)
- Une route aérienne



● Graphe orienté

- Graphe dans lequel toutes les arêtes sont orientées
- Réseau routier urbain (sens unique...)

● Graphe non-orienté

- Graphe dans lequel toutes les arêtes sont non-orientées
- Réseau des routes aériennes

Terminologie

● Extrémités d'une arête

- Les sommets U et V sont les extrémités de l'arête a

● Arêtes incidentes à un sommet

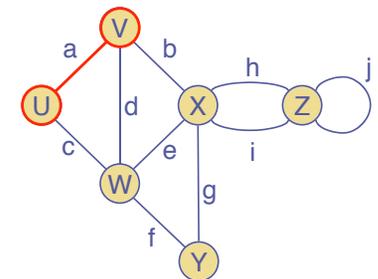
- Les arêtes a, b et d sont incidentes au sommet V

● Sommets adjacents

- Deux sommets sont adjacents s'ils sont reliés par une arête. Ex: les sommets U et V sont adjacents

● Degré d'un sommet

- Le degré d'un sommet est le nombre d'arêtes adjacentes à ce sommet. Ex: le degré du sommet V est 3.



● Arêtes multiples: h et i

● Boucle

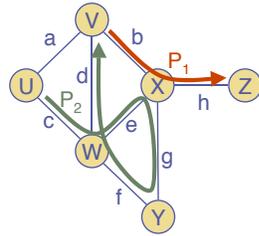
- Une boucle est une arête ayant deux fois le même sommet comme extrémité. Ex: j est une boucle

- Lorsqu'on compte le degré d'un sommet une boucle compte pour 2

Terminologie (suite)

● Chemin

- Séquence alternante de sommets et d'arêtes
- Commence dans un sommet
- Se termine dans un sommet
- Dans le séquence, chaque arête se trouve entre ses extrémités



● Chemin simple

- Chemin dans lequel tous les sommets et toutes les arêtes visités sont différents

● Exemples:

- $P_1=(V,b,X,h,Z)$ est un chemin simple
- $P_2=(U,c,W,e,X,g,Y,f,W,d,V)$ n'est pas un chemin simple

Terminologie (suite)

● Cycle

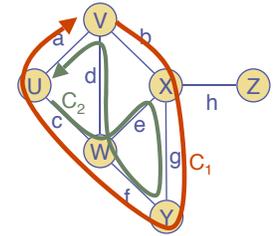
- Un cycle est un chemin commençant et se terminant dans le même sommet

● Cycle simple

- Cycle dans lequel tous les sommets et toutes les arêtes visités sont différents

● Exemples:

- $C_1=(V,b,X,g,Y,f,W,c,U,a,V)$ est un cycle simple
- $C_2=(U,c,W,e,X,g,Y,f,W,d,V,a,U)$ n'est pas un cycle simple



Quelques propriétés

Propriété 1:

$$\sum_{s \in S} \deg(s) = 2m$$

Preuve: chaque arête est comptée deux fois

Notations:

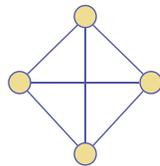
- n nombre de sommets
- m nombre d'arêtes
- $\deg(s)$ degré du sommet s

Propriété 2:

Dans un graphe non-orienté n'ayant aucune boucle ou arête multiple, on a

$$m \leq \frac{n(n-1)}{2}$$

Preuve: chaque sommet a un degré d'au plus $n-1$



Exemple

- $n = 4$
- $m = 6$
- $\deg(v) = 3$

Opérations principales du TAD graphe

Les sommets et les arêtes

- sont des positions
- garde en mémoire des éléments

Opérations

- **extrémités(e):** une liste des extrémité de e
- **opposé(v, e):** l'extrémité de e différente de v
- **sontAdjacente(v, w):** vrai ssi v et w sont adjacentes
- **remplace(v, x):** remplace l'élément du sommet v par x
- **remplace(e, x):** remplace l'élément de l'arête e avec x

Opérations de mise à jour

- **insérerSommet(o):** insère un sommet isolé contenant l'élément o
- **InsérerArête(v, w, o):** insère une arête (v,w) gardant en mémoire l'élément o
- **enlèveSommet(v):** enlève le sommet v (et les arêtes adjacentes)
- **enlèveArête(e):** enlève l'arête e

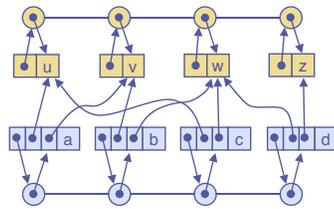
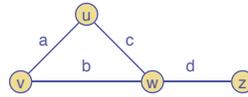
Opérations retournant un itérateur

- **incidentes(v):** retourne un itérateur des arêtes incidentes au sommet v
- **sommets():** retourne un itérateur des sommets du graphe
- **arêtes():** retourne un itérateur des arêtes du graphe

Structures de données pour les graphes

1. Liste d'arêtes

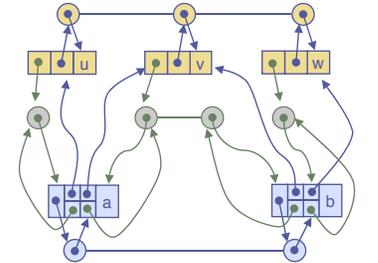
- un noeud S pour chaque sommet contenant
 - élément gardé en mémoire par le noeud
 - Pointeur vers la position du noeud dans la séquence de noeuds
- un noeud A pour chaque arête contenant
 - élément gardé en mémoire par l'arête
 - pointeur vers le noeud S contenant l'origine de l'arête
 - pointeur vers le noeud S contenant la destination de l'arête
 - pointeur vers la position du noeud dans la séquence de noeud A



Structures de données pour les graphes (suite)

2. Liste d'adjacence

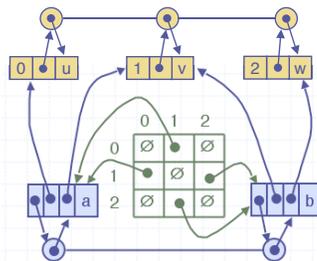
- généralise la structure liste d'arête dans le sens suivant:
 - On a une nouvelle séquence, la séquence d'indidence et chaque noeud S a maintenant un pointeur vers les arêtes incidentes
 - chaque noeud A a maintenant des pointeurs vers les positions de ses extrémités dans la séquence d'adjacence



Structures de données pour les graphes (suite)

3. Matrice d'adjacence

- généralise la structure liste d'arête dans le sens suivant:
 - Chaque noeud S garde en mémoire un entier désignant la position du sommet dans la séquence
 - Matrice d'adjacence. En (i,j) pointeur vers l'arête reliant les sommets i et j ou \emptyset si les sommets ne sont pas adjacents



Performance

n sommets, m arêtes aucune arête-multiple aucune boucle	Liste Arêtes	Liste d'adjacence	Matrice d'adjacence
Espace	$n + m$	$n + m$	n^2
incidences(v)	m	$\text{deg}(v)$	n
sontAdjacent(v, w)	m	$\min(\text{deg}(v), \text{deg}(w))$	1
insèreSommet(a)	1	1	n^2
insèreArête(v, w, a)	1	1	1
enlèveSommet(v)	m	$\text{deg}(v)$	n^2
enlèveArête(e)	1	1	1