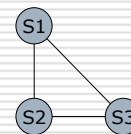


## Algorithmes complexes Partie 3

### Les graphes

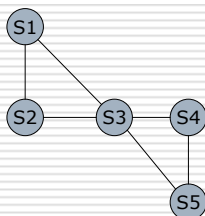
### Définition

- Un graphe est une structure de données composée d'un ensemble de sommets, et d'un ensemble de relations entre ces sommets.
- Si la relation n'est pas orientée, la relation est supposée exister dans les deux sens. Le graphe est dit non orienté ou symétrique.
- Si les relations sont orientées, le graphe est dit orienté.
- Une relation est appelée arc (ou arête pour les graphes non orientés).
- Les sommets sont aussi appelés nœuds ou points.



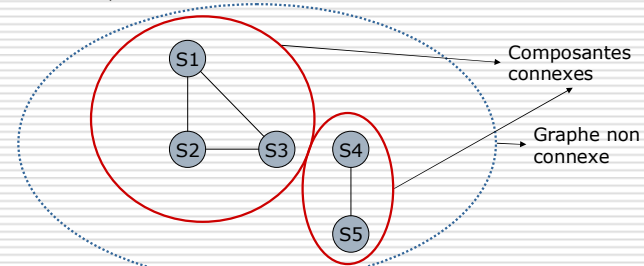
### Graphes non orientés

- Ensemble des sommets :  
 $S = \{S1, S2, S3, S4, S5\}$
- Ensemble des relations symétriques  
 $A = \{S1S2, S1S3, S2S3, S3S4, S3S5, S4S5\}$



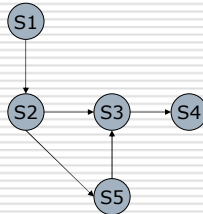
### Graphes connexes

- Un graphe non orienté est dit connexe si on peut aller de tous les sommets vers tous les autres sommets.
- En suivant cette définition, un graphe peut ne pas être connexe mais avoir des composantes connexes.



### Graphes orientés

- ❑ Si les relations sont orientées, le graphe est dit orienté.
- ❑ S2 est le successeur de S1.
- ❑ S3 est le prédécesseur de S4.

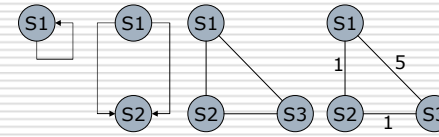


Algorithmes complexes : Les graphes

5

### Propriétés d'un graphe

- ❑ Une **boucle** (autoboucle) est une relation (Si, Si).
- ❑ Un **multigraphe** est un graphe tel qu'il existe plusieurs arcs (allant dans le même sens) entre certains sommets.
- ❑ Un **graphe simple** est un graphe sans boucle et sans arc multiple.
- ❑ Un graphe est dit **pondéré** si à chaque arc est associé une valeur représentant le coût de transition de cet arc.
- ❑ Un **chemin** est une suite d'arcs consécutifs.
- ❑ La **longueur d'un chemin** représente le nombre d'arcs composant ce chemin.

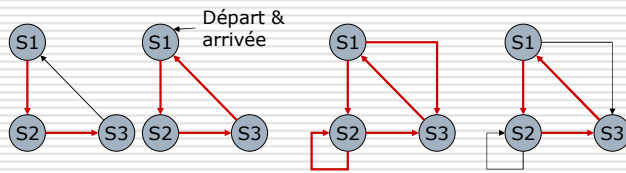


Algorithmes complexes : Les graphes

6

### Propriétés d'un graphe (suite)

- ❑ Un **chemin simple** est un chemin où aucun arc n'est utilisé 2 fois.
- ❑ Un **circuit simple** est un chemin simple dont le sommet de départ et le sommet d'arrivée sont les mêmes.
- ❑ Un **circuit eulérien** est un circuit simple qui passe une fois et une seule fois par tous les arcs.
- ❑ Un **circuit hamiltonien** est un circuit qui passe une fois et une seule fois par tous les sommets.

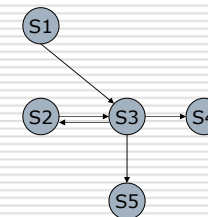


Algorithmes complexes : Les graphes

7

### Degré d'un graphe

- ❑  $D^{\circ}(S3) = 5$  : Degré du sommet S3, nombre d'arcs entrants ou sortants
- ❑  $D^{+}(S3) = 3$  : Degré d'émission, nombre d'arcs sortants
- ❑  $D^{-}(S3) = 2$  : Degré de réception, nombre d'arcs entrants



Algorithmes complexes : Les graphes

8

## Spécification abstraite

**Sorte** graphe

**Utilise** entier, sommet, arc, booléen, élément, liste

### Opérations:

Créer\_graphe :  $\rightarrow$  graphe

Création d'un graphe vide

Sommet : graphe  $\otimes$  entier  $\rightarrow$  Sommet

Retourne le nième sommet

Arcs : graphe  $\otimes$  entier  $\rightarrow$  Liste d'arcs

Retourne la liste des arcs sortants du nième sommet

Marquer\_sommet : graphe  $\otimes$  entier  $\rightarrow$  graphe

Permet de marquer un sommet

Est\_marqué : graphe  $\otimes$  entier  $\rightarrow$  booléen

Teste le marquage d'un sommet

## Spécification abstraite (suite)

Nb\_sommet : graphe  $\rightarrow$  entier

Retourne le nombre de sommet d'un graphe

razMarque : graphe  $\rightarrow$  graphe

Supprime tous les marquages d'un graphe

Ajouter\_sommet : graphe  $\otimes$  élément  $\rightarrow$  graphe

Ajout un élément dans un graphe

Ajouter\_arc : graphe  $\otimes$  sommet  $\otimes$  sommet  $\otimes$  entier  $\rightarrow$  graphe

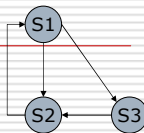
Ajout un arc entre deux sommets

### Fin\_sorte

## Mémorisation sous forme de matrice d'adjacence

□ Un vecteur de taille max n pour les sommets

□ Une matrice de taille n\*n pour les arcs



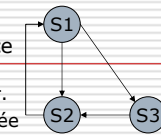
	Nom Sommet
0	S1
1	S2
2	S3
...	
N-1	

	0	1	2	...	N-1
0		v	v		
1	v				
2		v			
...					
N-1					

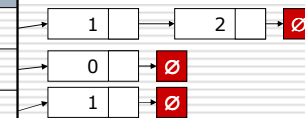
## Mémorisation en table de listes d'adjacence

□ Les sommets sont stockés dans un vecteur.

□ Les arcs sont stockés dans une liste associée aux sommets.

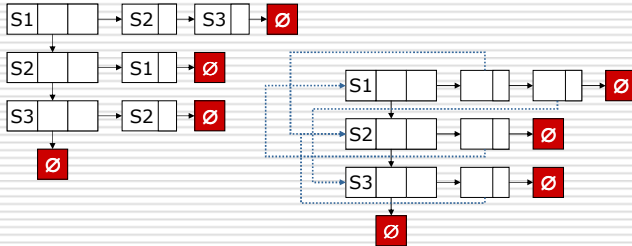


	Nom Sommet	Li
0	S1	
1	S2	
2	S3	
...		
N-1		



### Allocation dynamique

- Les sommets sont stockés dans une liste.
- Les arcs sont représentés :
  - par le nom du sommet vers lequel il pointe.
  - ou par un pointeur vers ce sommet.

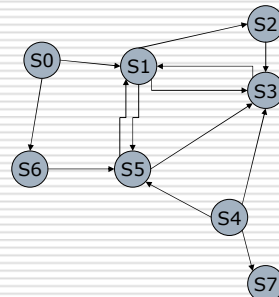


### Parcours en profondeur

1. On choisit un premier sommet qu'on empile.
  2. Ce sommet est marqué comme visité.
  3. A partir de ce sommet, on empile un sommet fils qui n'a pas été visité.
  4. Ce sommet est marqué comme visité.
  5. A partir de ce nouveau sommet, on empile un sommet fils qui n'a pas été visité.
  6. Ce sommet est marqué comme visité.
  7. Ainsi de suite...
  8. Si on arrive dans une impasse, on dépile les sommets jusqu'à trouver un sommet ayant un fils non-marqué.
  9. Si on trouve un fils, celui-ci est empilé et marqué.
  10. Sinon la pile va se vider complètement. Auquel cas on empile un nouveau sommet parmi ceux non marqué.
  11. On recommence le processus à partir du point 3.
  12. Une fois la pile vide et tous les éléments marqués, le parcours du graphe est terminé.
- Remarque : Le traitement effectué lors du parcours (par exemple l'affichage d'un sommet) s'effectue lorsqu'un sommet est marqué.

### Exemple de parcours en profondeur

- S0
- S1
- S2
- S3
- S5
- S6
- S4
- S7



### Parcours en largeur

- Pour le parcours en largeur, on utilise une file de sommets.
1. On choisit un premier sommet qu'on inclut dans la file.
  2. A partir de ce sommet, on met dans la file tous ses fils non marqués.
  3. On supprime ce sommet de la file et il est marqué comme visité.
  4. On sélectionne le prochain sommet de la file.
  5. A partir de ce nouveau sommet, on met dans la file tous ses fils.
  6. On supprime ce sommet de la file et il est marqué comme visité.
  7. Ainsi de suite...
  8. La file va se vider complètement. Auquel cas on ajoute un nouveau sommet parmi ceux non marqué.
  9. On recommence le processus à partir du point 3.
  10. Une fois la pile vide et tous les éléments marqués, le parcours du graphe est terminé.
- Remarque : Le traitement effectué lors du parcours (par exemple l'affichage du graphe) s'effectue lorsqu'un sommet est marqué.

Exemple de parcours en largeur

S0 S1 S6 S2 S3 S5  
S4 S7

