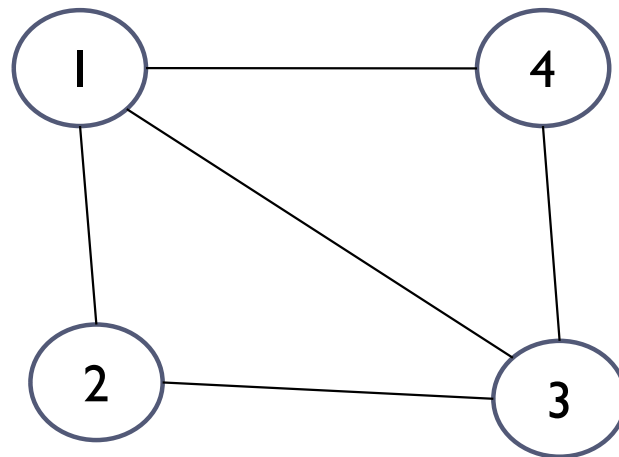


4.3. Les Graphes - Définition

- ▶ Un graphe **G** est constitué de 2 ensembles **V** et **E** .
- ▶ **V** est un ensemble non-vide de sommets.
- ▶ **E** est un ensemble de paires de sommets de **V** .
 - **$E = \{ (v, v') / v, v' \in V \}$**
 - Ces paires sont appelés arêtes.
- ▶ On écrit aussi **$G = (V, E)$** pour représenter le graphe.

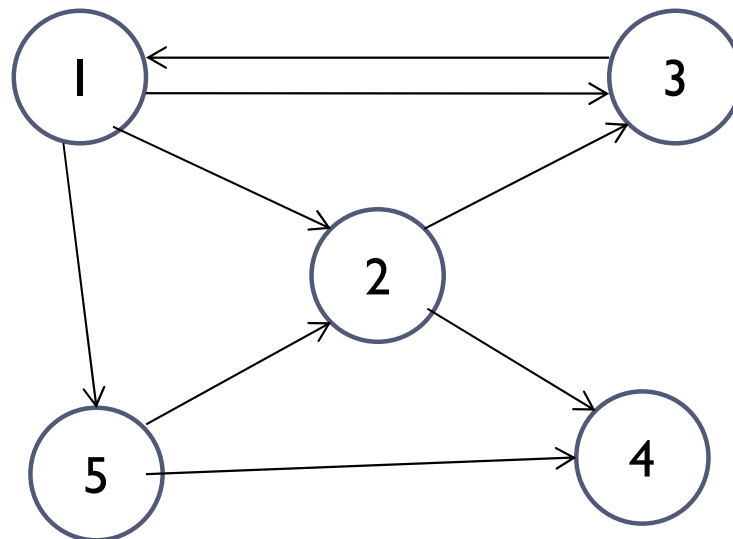
4.3. Les Graphes - Terminologie

- ▶ **Graphe non orienté** : la paire de sommets représentant une arête n'est pas ordonnée.
 - ▶ Autrement dit, les (v_1, v_2) et (v_2, v_1) représentent la même arête.



4.3. Les Graphes - Terminologie

- ▶ **Graphe orienté** : Chaque arête est représentée par la paire orientée (v_1, v_2) ; v_1 étant l'origine et v_2 l'extrémité de l'arête.
- ▶ Dans ce cas, on ne parle plus d'arête mais d'arc. Par conséquent, (v_2, v_1) et (v_1, v_2) représentent deux arcs différents.



4.3. Les Graphes -Terminologie

- ▶ **Adjacence** : deux sommets sont adjacents s'il existe un arc, ou une arête, les reliant.
- ▶ **Arc** : couple (x,y) dans un graphe orienté.
- ▶ **Arête** : couple (x,y) dans un graphe non orienté.
- ▶ **Chemin** : suite d'arcs connexes reliant un sommet à un autre dans un graphe orienté. Par exemple $(a;b) (b;c) (c;d) (d;b) (b;e)$ est un chemin reliant **a** à **e** ; on le note (a,b,c,d,b,e) .
- ▶ **Degré d'un sommet** : nombre d'arêtes issues d'un sommet dans un graphe non orienté ; nombre d'arcs arrivant ou partant d'un sommet dans un arc orienté.
- ▶ **Prédécesseur - Successeur** : Dans l'arc $(x;y)$, **x** est prédécesseur de **y** qui est le successeur de **x**

Représentation Statique d'un graphe

Graphe = enreg

nb_sommet : entier

sommet : tableau[1..Max] de valeur

relation : tableau[1..Max, 1..Max] de booléen

fenreg

Représentation Dynamique d'un graphe

▶ **Graphe = ^somet**

▶ **somet = enreg**

val : valeur

successeur : list_succ

next : Graphe

fenreg

▶ **List_succ = ^succ**

▶ **succ = enreg**

somet_succ : Graphe

next_succ : list_succ

fenreg

Application

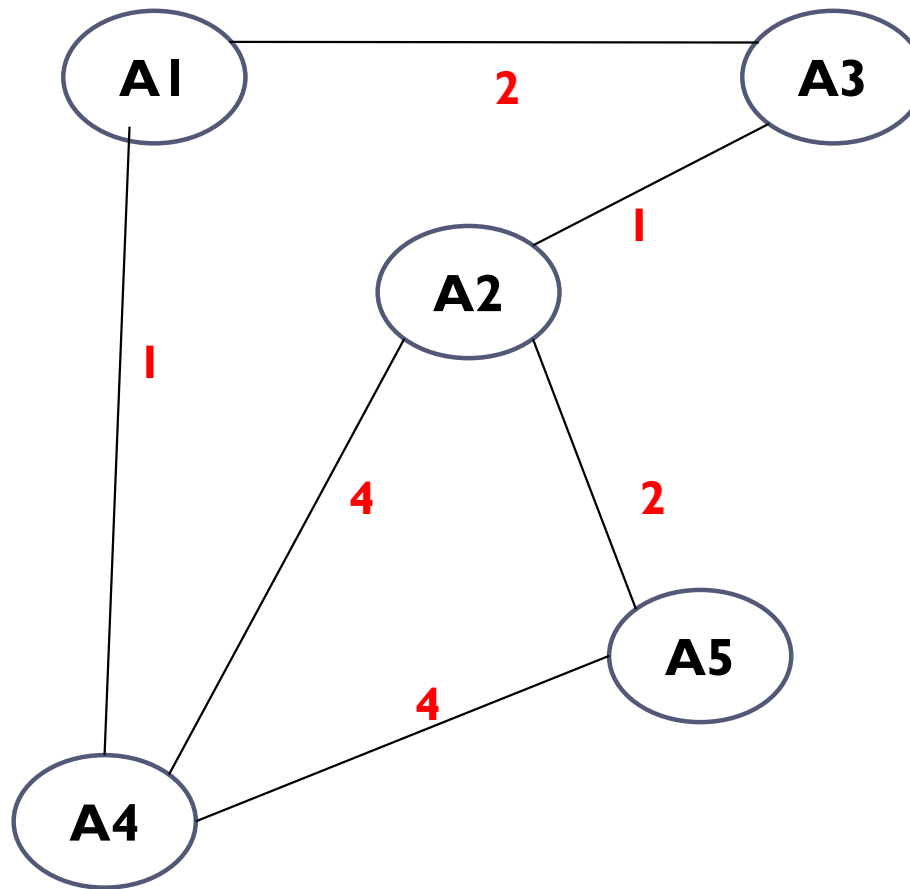
- ▶ Une société de production cinématographique désire informatiser son système de gestion. Ainsi, on vous a confié la conception du module « gestion des acteurs ». Pour ce faire, on dispose d'un ensemble de N acteurs. Chaque acteur est caractérisé par les informations suivantes :
 - numéro d'identification (entier)
 - nom (chaîne de 30 caractères)
 - adresse (chaîne de 50 caractères)
 - sexe (1 caractère : M : masculin, F : féminin)
 - nationalité (chaîne de 30 caractères)
- ▶ De plus, pour chaque acteur, on connaît ses partenaires (les acteurs ayant joué avec lui) et le nombre de films auxquels il avait participé avec chacun d'eux.
- ▶ **Exemple :**
Soit l'ensemble d'acteurs : $\{A1, A2, A3, A4, A5\}$.
On sait que:
 - ▶ A1 a joué avec A3 dans 2 films et avec A4 dans 1 film.
 - ▶ A2 a joué avec A3 dans 1 film, avec A4 dans 4 films et avec A5 dans 2 films
 - ▶ A4 et A5 ont joués ensemble dans 4 films.

I) Quelle structure abstraite utilisera-t-on pour modéliser ce problème?

Application

On utilisera un graphe

Exemple :



Application

2) Représenter ce graphe en utilisant :

a) Une structure de données statique RI

acteur = enreg

num : entier

nom : chaîne

adresse : chaîne

sexe : caractère

nationalité : chaîne

fenreg

Graphe = enreg

nb_sommet : entier

sommet : tableau[1..Max] de acteur

relation : tableau[1..Max, 1..Max] d'entier

fenreg

Application

b) Une structure de données dynamique R2.

Graphe = ^somet

somet = enreg

val : acteur

successeur : list_succ

next : Graphe

fenreg

List_succ = ^succ

succ = enreg

somet_succ : Graphe

nb_films : entier

next_succ : list_succ

fenreg

Application

- 3) En utilisant la structure de données RI, écrire une procédure qui affiche les noms des **actrices** ayant jouées avec un acteur de numéro d'identification donné.

Application

- 3) En utilisant la structure de données RI, écrire une procédure qui affiche les noms des **actrices** ayant jouées avec un acteur de numéro d'identification donné.

Procédure actrices (G: Graphe ; num : entier)

Var pos, j : entier

Début

pos := get_pos (G , num)

Si pos \neq 0 Alors

pour j de 1 à G.nb_sommet faire

Si G.relation[pos,j] \neq 0 et G.sommet[j].sexe == 'F' alors
écrire(G.sommet[j].nom)

fsi

fpour

fsi

Fin

Application

3) En utilisant la structure de données RI, écrire une procédure qui affiche les noms des actrices ayant jouées avec un acteur de numéro d'identification donné.

Fonction get_pos (G: Graphe ; num : entier) : entier

Var i : entier

Début

i := 1

tant que i ≤ G.nb_sommet et G.sommet[i].num <> num faire

i := i + 1

fait

Si i ≤ G.nb_sommet Alors

get_pos := i

Sinon

get_pos := 0

fsi

Fin

Application

- 4) En utilisant la structure de données RI, écrire une fonction qui calcule le nombre de films joués par un acteur de numéro d'identification donné.

Application

- 4) En utilisant la structure de données RI, écrire une fonction qui calcule le nombre de films joués par un acteur de numéro d'identification donné.

Fonction nb_films (G: Graphe ; num : entier) : entier

Var n, pos, j : entier

Début

 n := 0

 pos := get_pos(G , num)

 Si pos <> 0 Alors

 pour j de 1 à G.nb_sommet faire

 n := n + G.relation[pos,j]

 fpour

 fsi

 nb_films := n

Fin

Application

4) En utilisant la structure de données R2, écrire :

- a) Une procédure qui retourne dans une structure de votre choix l'ensemble de tous les acteurs (avec les informations qui les caractérisent) et pour chaque acteur uniquement les partenaires portant la même nationalité que lui.

Application

4) En utilisant la structure de données R2, écrire :

a) Une procédure qui retourne dans une structure de votre choix l'ensemble de tous les acteurs (avec les informations qui les caractérisent) et pour chaque acteur uniquement les partenaires portant la même nationalité que lui.

b) Une procédure qui affiche les noms des acteurs qui ont joué à la fois avec deux acteurs de numéros donnés.

Pour l'exemple précédent et si les acteurs donnés sont A1 et A2, la procédure affichera A3 et A4.