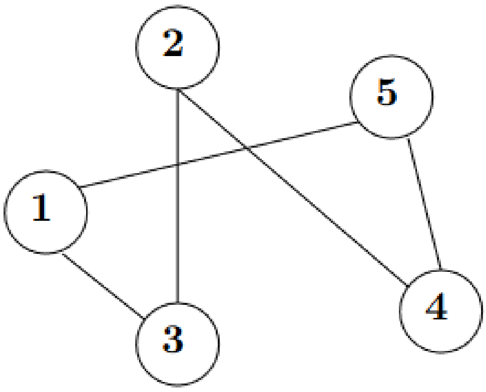
Graphes et algorithmes : TD2

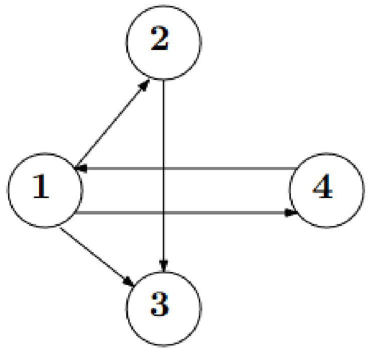
Exercice 1

Sujet

Donner les représentations par matrice d'incidence, matrice d'adjacence et listes d'adjacence des deux graphes suivants, puis déterminer le degré de chaque sommet.



1



2

Résolution

Graphe 1 :

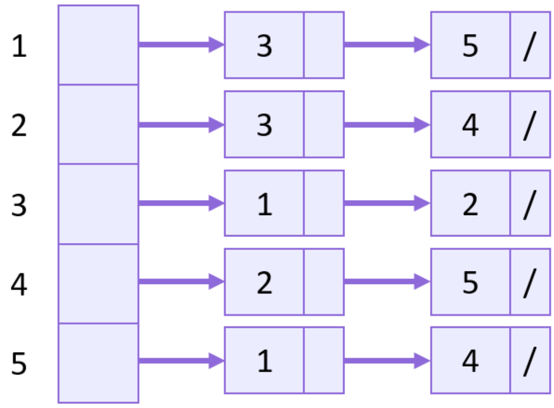
- **Matrice d'incidence** :

$$
\begin{array}{c c}
 & \begin{array}{ccccc} (1,3) & (1,5) & (2,3) & (2,4) & (4,5) \end{array} \\
\begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} & \begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 \end{pmatrix}
\end{array}
$$

- **Matrice d'adjacence** :

$$
\begin{array}{c c}
 & \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\
\begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} & \begin{pmatrix} 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \end{pmatrix}
\end{array}
$$

- **Listes d'adjacence** :



Graphe 2 :

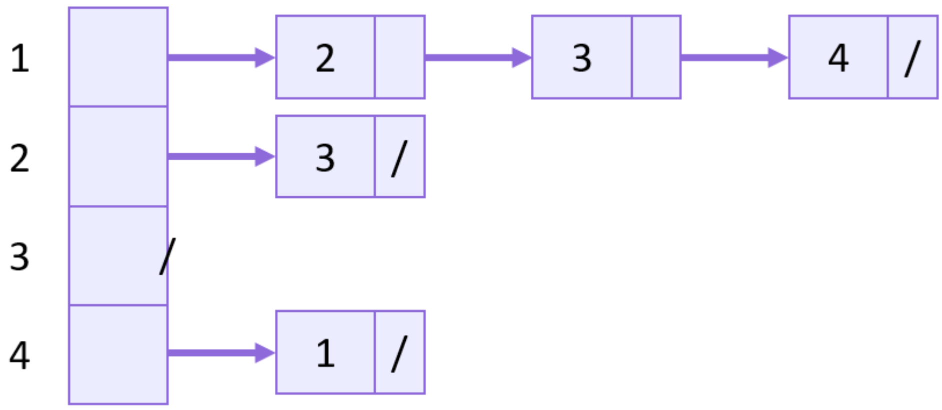
- **Matrice d'incidence** :

$$\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{ccccc} (1,2) & (1,3) & (1,4) & (2,3) & (4,1) \end{array} \\ \begin{pmatrix} 1 & 1 & 1 & 0 & -1 \\ -1 & 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & -1 & 0 \\ 0 & 0 & -1 & 0 & 1 \end{pmatrix} \end{array}$$

- **Matrice d'adjacence** :

$$\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{pmatrix} 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} \end{array}$$

- **Listes d'adjacence** :



Exercice 2

Sujet

Donner des algorithmes (en pseudo-langage) pour insérer et supprimer un arc dans un graphe orienté représenté par sa matrice d'adjacence, puis par listes d'adjacences.

Résolution

Matrice d’adjacence

```
CONST MAX = 1000;
TYPE Matrice = tableau[MAX][MAX] d'Entiers;

/*INSERTION*/

/*retourne 0 si (i,j) existe déjà, 1 sinon*/
fonction ajout_matrice(Entrée/Sortie M : Matrice, Entrée i,j : Entier) : En
début
    si M[i][j] = 0 alors
        M[i][j] ← 1;
        retourner 1;
    fin si
    retourner 0;
fin

/*SUPPRESSION*/

/*retourne 0 si (i,j) n'existe pas, 1 sinon*/
fonction supp_matrice(Entrée/Sortie M : Matrice, Entrée i,j : Entier) : Ent
début
    si M[i][j] = 1 alors
        M[i][j] ← 0;
        retourner 1;
    fin si
    retourner 0;
fin
```

Listes d’adjacence

```
CONST MAX = 1000;
TYPE Listes = tableau[MAX] de Liste;

/*INSERTION*/

/*retourne 0 si (i,j) existe déjà, 1 sinon*/
fonction ajout_liste(Entrée/Sortie L : Listes, Entrée i,j : Entier) : Entie
début
    si j ∉ L[i] alors
        L[i] ← L[i] + {j};
        retourner 1;
    fin si
    retourner 0;
```

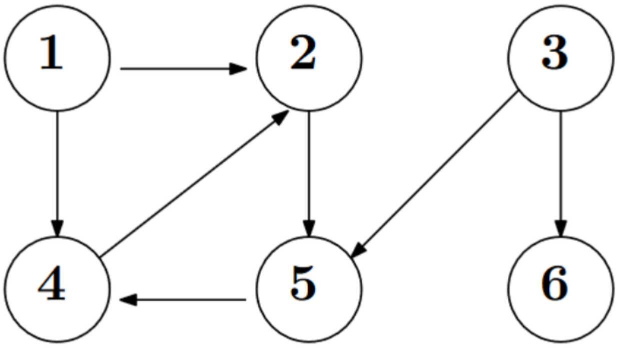
```
fin

/*SUPPRESSION*/

/*retourne 0 si (i,j) n'existe pas, 1 sinon*/
fonction supp_liste(Entrée/Sortie L : Listes, Entrée i,j : Entier) : Entier
début
    si j ∈ L[i] alors
        L[i] ← L[i] - {j};
        retourner 1;
    fin si
    retourner 0;
fin
```

Exercice 3

Sujet

Le carré d'un graphe orienté $G = (S, A)$ est le graphe $G^2 = (S, A^2)$ tel que $(x, z) \in A2 \Leftrightarrow \exists y \in S$ tel que $(x, y) \in A$ et $(y, z) \in A$.

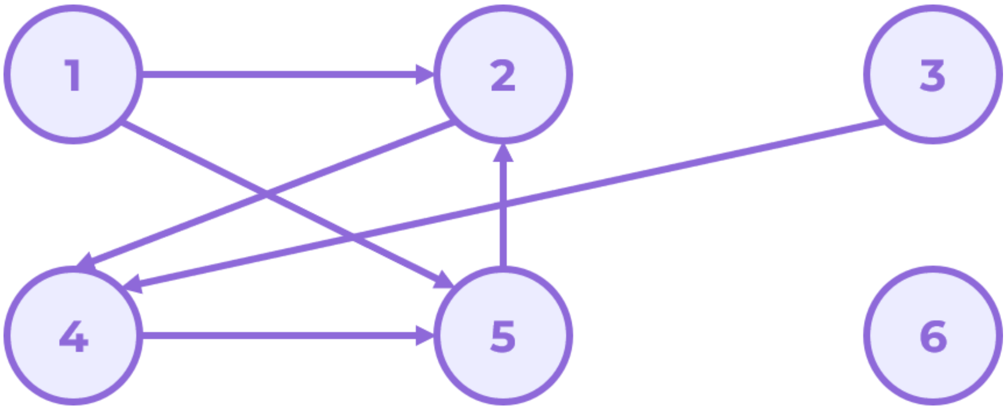
1. Donner le carré du graphe orienté ci-dessous :



2. Donner une interprétation de G^2 en termes de chemin.
3. Décrire un algorithme (pseudo-langage) efficace permettant de calculer le graphe G^2 d'un graphe G représenté par listes d'adjacence.
4. Même question si G est représenté par matrice d'adjacence.
5. Analyser le temps d'exécution des algorithmes décrits dans les questions précédentes.

Résolution

Question 1



Question 2

Pour que l'arc (x, z) existe dans G^2, il faut qu'il existe au moins un chemin de longueur 2 dans G reliant x à z.

Question 3

```
/*retourne le carré du graphe représenté par la liste
d'adjacence L*/
fonction carre_listes(Entrée L : Listes, n : Entier) : Listes
début
    pour i allant de 0 à n faire
        L²[i] ← ∅;
    fin pour

    pour i allant de 0 à n faire
        pour j ∈ L[i] faire
            pour k ∈ L[j] faire
                ajout_liste(L², i, k);
            fin pour
        fin pour
    fin pour
    retourner L²;
fin
```

> Complexité : $O(n^3)$

Exercice 4

Sujet

1. Donner un algorithme qui transforme la représentation d'un graphe par la liste des successeurs en une représentation par la liste des prédécesseurs. Analyser sa complexité.

2. Donner deux algorithmes, un qui convertit un graphe représenté par sa matrice d'adjacence en liste d'adjacence et l'autre qui effectue l'opération inverse. Évaluer leur complexité.

3. Donner un algorithme qui calcule et retourne le plus petit degré de tous les sommets d'un graphe orienté, s'il est représenté par sa matrice d'adjacence, puis par ses listes d'adjacence. Évaluer leur complexité.

Résolution

Question 1

```
fonction liste_pred(Entrée L : Liste) : Liste

variables
    Lp : Liste;
    i, j : Entier;

début
    pour i allant de 1 à taille(L) faire
        Lp[i] ← ∅;
    fin pour

    pour i allant de 1 à taille(L) faire
        pour j ∈ L[i] faire
            Lp[j].ajoute(i);
    retourner Lp;
fin
```

> Complexité : $O(n^2)$

Question 2

```
/* retourne la matrice d'adjacence du graphe représenté par la liste
d'adjacence L */

fonction matrice_adj(Entrée L : Liste) : Matrice

variables
    M : Matrice;
    i, j : Entier;

début
    pour i allant de 1 à taille(L) faire
        pour j allant de 1 à taille(L) faire
            M[i][j] ← 0;
```

```
        fin pour
    fin pour

    pour i allant de 1 à taille(L) faire
        pour j ∈ L[i] faire
            M[i][j] ← 1;
        fin pour
    fin pour
    retourner M;
fin
```

Complexité : $O(n^2)$

```
/* retourne la liste d'adjacence du graphe représenté par la matrice
d'adjacence M */

fonction liste_adj(Entrée M : Matrice) : Liste

variables
    L : Liste;
    i, j : Entier;

début
    pour i allant de 1 à taille(M) faire
        L[i] ← ∅;
    fin pour

    pour i allant de 1 à taille(M) faire
        pour j allant de 1 à taille(M) faire
            si M[i][j] = 1 alors
                L[i].ajoute(j);
            fin si
        fin pour
    fin pour
    retourner L;
fin
```

Complexité : $O(n^2)$

Question 3

```
fonction degre_min_matrice(Entrée M : Matrice) : Entier

variables
```

```
    i, j, d, min : Entier;

début
    min ← taille(M);
    pour i allant de 1 à taille(M) faire
        d ← 0;
        pour j allant de 1 à taille(M) faire
            d ← d + M[i][j] + M[j][i];
        fin pour
        d ← d - M[i][i];
        si d < min alors
            min ← d;
        fin si
    fin pour
    retourner min;
fin
```

> Complexité : $O(n^2)$

$\neq$

```
fonction degre_min_liste(Entrée L : Liste) : Entier

variables
    i, j, d, min : Entier;

début
    min ← taille(L);
    pour i allant de 1 à taille(L) faire
        d ← 0;
        pour j ∈ L[i] faire
            d ← d + 1;
        fin pour
        pour j allant de 1 à taille(L) faire
            si j != i alors
                pour k ∈ L[j] faire
                    si k = i alors
                        d ← d + 1;
                    fin si
                fin pour
            fin si
        fin pour
        si d < min alors
            min ← d;
        fin si
    fin pour
```

```
    retourner min;
fin
```

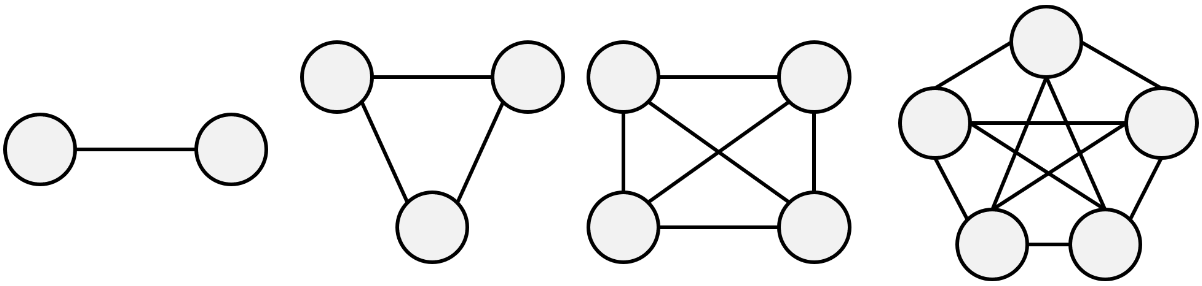
> Complexité : $O(n^3)$

Exercice 6

Sujet

Dessiner les graphes (non orientés) complets d'ordre 2, 3, 4 et 5. Préciser pour chacun le degré de chaque sommet et le nombre total d'arêtes.

Résolution



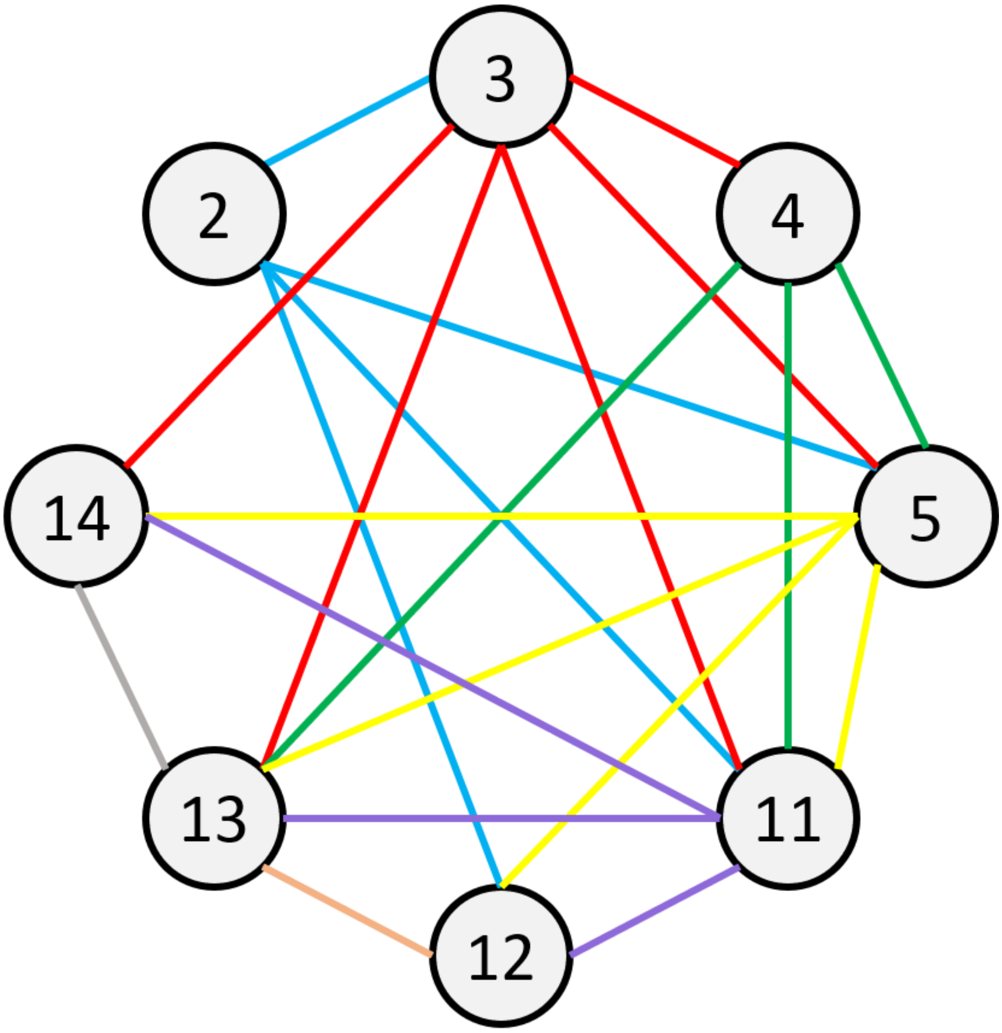
- Ordre 2 : 1 arête, degré 1 pour chaque sommet.
- Ordre 3 : 3 arêtes, degré 2 pour chaque sommet.
- Ordre 4 : 6 arêtes, degré 3 pour chaque sommet.
- Ordre 5 : 10 arêtes, degré 4 pour chaque sommet.

Exercice 8

Sujet

Définir un graphe non orienté $G = (S, A)$ tel que $S = \{2, 3, 4, 5, 11, 12, 13, 14\}$ et tel que deux sommets i et j sont adjacents si et seulement si $\text{pgcd}(i, j) = 1$ (pgcd $=$ plus grand diviseur commun). Quelles est le nombre d'arêtes de G ?

Résolution



Il y a 21 arêtes.

Exercice 9

Sujet

Soit $G = (S, A)$ un graphe non orienté comprenant 8 sommets et 15 arêtes. Tous les sommets de G sont de degré 3 ou 5. Combien de sommets de degré 3 et 5 a le graphe G ? (justifier la réponse) Construire un exemple pour G.

Résolution

Posons a le nombre de sommets de degré 3 et b le nombre de sommets de degré 5. On a :

$$\begin{cases} a + b = 8 \\ 3a + 5b = 30 (15 \times 2) \end{cases}$$

On résout le système :

$$\begin{cases} a = 5 \\ b = 3 \end{cases}$$