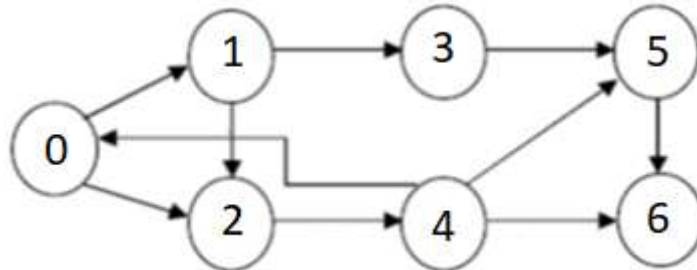


## FICHE TD N°5 LES GRAPHS

**Exercice 1 :**

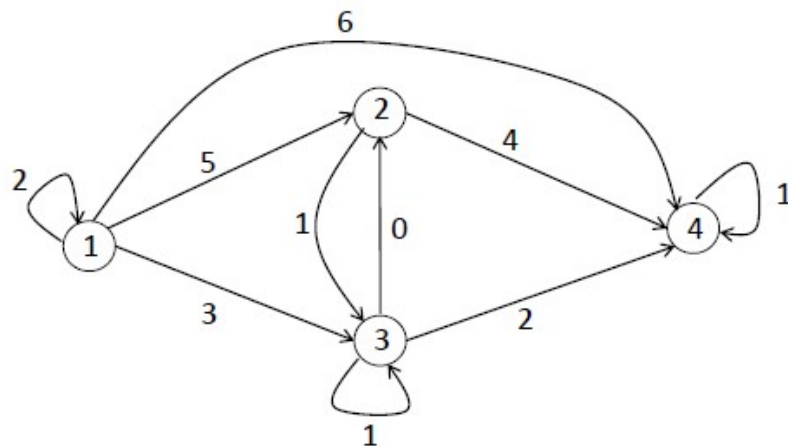
Soit le graphe orienté suivant :



1. Calculer sa densité
2. Donner la représentation du graphe sous forme de matrice d'adjacence puis sous forme de listes d'adjacences (chaînées et contiguës) et analyser la complexité spatiale de chaque représentation.
3. Ecrire pour chaque représentation l'algorithme permettant de tester si un deux sommets donnés  $u$  et  $v$  sont voisins puis analyser la complexité temporelle de ces algorithmes.

**Exercice 2 :**

Soit le graphe suivant :



1. Dérouler l'algorithme de parcours BFS à partir du sommet 1.
2. Dérouler l'algorithme de parcours DFS à partir du sommet 1.
3. Indiquer la complexité de ces algorithmes en fonction du nombre de sommets  $n$  et du nombre d'arcs  $m$ .

**Exercice 3 :**

Soit un graphe  $G$  orienté non valué représenté en matrice d'adjacence avec la déclaration en C suivante :

```

#define N 999
typedef struct g{
    int nbSommets, nbArcs;
    int Mat[N][N];
} Graphe;
  
```

Donnez les fonctions puis leurs complexités et qui permettent de :

- Retourner le graphe complémentaire d'un graphe.
- Modifier le graphe en entrée en éliminant toutes les boucles (doit être récursive).

#### **Exercice 4 :**

Un sommet source est un sommet qui n'a pas de prédécesseurs. Etant donné un graphe orienté de  $n$  sommets et  $m$  arcs représenté en listes chaînées d'adjacence, écrire une fonction booléenne permettant de tester si un sommet donné en paramètre est un sommet source. Donner les instructions de déclaration et analyser la complexité de votre fonction.

#### **Exercice 5 :**

Appliquer l'algorithme de Dijkstra sur le graphe de l'exercice 2 à partir du sommet 1.

#### **Exercice 6 :**

Une variante de la structure de données « liste d'adjacence » permet de représenter l'ensemble des sommets par une liste chaînée, où on associe à chaque sommet un pointeur vers son successeur qui appartient à la liste chaînée des sommets, comme dans l'exemple suivant. Déclarer cette structure puis écrire la fonction ajoutArc(u,v).

