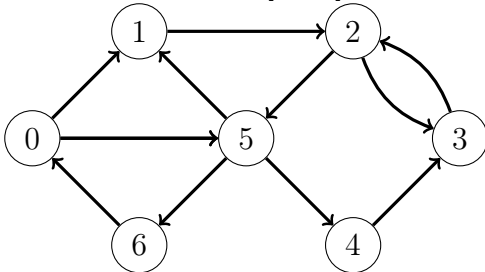


---

## TD 24 : GRAPHE : INTRODUCTION

---

On considère le graphe ci-dessous. On peut toujours construire une bijection entre les  $N$  sommets et l'ensemble  $[0, N[$ , on confond donc un sommet avec son index.



### Exercice 1 (Matrice d'adjacence)

Une première manière d'implémenter un graphe en informatique est d'utiliser sa matrice d'adjacence : elle est initialisée à 0, puis on représente par 1 une arête allant du sommet de la ligne vers le sommet de la colonne.

(1) Ecrire une fonction `init_matrice_adjacence(aretes)` où `aretes` est un tableau contenant un nombre pair de sommet, qui renvoie la matrice d'adjacence du graphe.

`aretes=[0,1,0,5]` signifie qu'il y a une arête qui part du sommet 0 pour aller au sommet 1, puis une arête qui part du sommet 0 pour aller au sommet 5, ...

(2) On écrira le tableau `aretes` qui correspond au graphe donné ci-dessus.

### Exercice 2 (Tests sur la matrice d'adjacence)

```
ma = init_matrice_adjacence(aretes)
```

(3) Ecrire une fonction `get_info_matrice(ma)` qui prend en argument une matrice d'adjacence et renvoie le tuple constitué d'un booléen (`True` si le graphe est orienté, `False` sinon), de l'ordre du graphe et de sa taille). Ainsi le tuple (`True, 7, 11`) correspond à un graphe orienté ayant 7 sommets et 11 arcs.

(4) Ecrire une fonction `voisins_matrice(ma, sommet)` qui renvoie un tuple contenant la liste des sommets voisins de `sommet` dans le graphe représenté par la matrice d'adjacence `ma`.

### Exercice 3 (Liste d'adjacence)

Une deuxième manière d'implémenter un graphe est d'utiliser une liste d'adjacence. Par exemple la liste d'adjacence `[[1,5],[2],...]` signifie que le sommet 0 est relié aux sommets 1 et 5, que le sommet 1 est relié au sommet 2, ...

(5) Ecrire une fonction `init_liste_adjacence(aretes)` qui renvoie la liste d'adjacence du graphe à partir du tableau `aretes` précédemment défini.

(6) Ecrire une fonction `get_info_liste(la)` qui prend en argument une liste d'adjacence et renvoie le tuple constitué d'un booléen (`True` si le graphe est orienté, `False` sinon), de l'ordre du graphe et de sa taille).