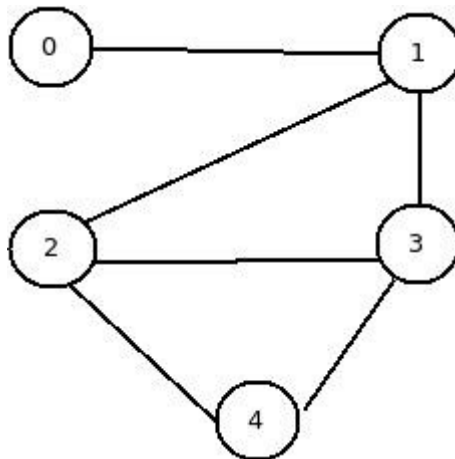


# TP d'Algorithmique Avancée

## Coloration de graphe

Soit le graphe non orienté  $G$ , suivant :



On rappelle l'algorithme de coloration gloutonne de graphe.

```
FONCTION Greedy_Coloring(  $G = (S, A)$  ) :
___ color = un dictionnaire vérifiant :  $\forall s \in S, \text{color}[s] = -1$ 
___ POUR  $i = 0, \dots, n-1$  :
___   # ensemble des couleurs des sommets adjacents au sommet  $i$ 
___    $C_i = \{\text{color}[j] : (i, j) \text{ est une arête}\}$ 
___   # le plus petit entier positif n'appartenant pas à  $C_i$ 
___    $\text{color}[i] = \min \{c \in \mathbb{N} : c \notin C_i\}$ 
___ FIN POUR
___ RETOURNER color
```

## Exercice 1

Implémenter le graphe  $G$  via l'utilisation d'une matrice d'adjacence.

On complétera le code suivant.

```
import numpy

def matrice(n):
    return numpy.zeros((n, n))

# Représentation par matrice d'adjacence
m = matrice(5)
m[0, 1] = 1
m[1, 0] = 1
```

## Exercice 2

Implémenter une fonction *succ*, prenant en argument un graphe (vu comme une matrice d'adjacence), ainsi qu'un certain sommet  $s$  (vu comme un entier) ; et retournant l'ensemble des sommets adjacents à  $s$ .

## Exercice 3

Implémenter la coloration gloutonne de graphe ; puis l'exécuter sur le graphe  $G$ .