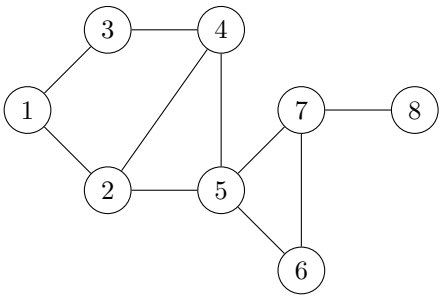
Graphes et Algorithmes

Travaux Pratiques n°1

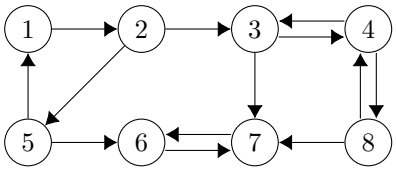
Université de la Nouvelle-Calédonie, 2024

Introduction

Durant ce TP, nous allons – enfin, vous allez – chercher à définir et manipuler un graphe. Dans la théorie des graphes, on retrouve deux types de graphes : les graphes non orientés et les graphes orientés. Les arêtes des graphes peuvent être associées à des valeurs. Si aucune valeur n'est affichée, on considérera que sa valeur vaut 1 (pour représenter la présence d'un lien entre deux sommets).



Graphe non orienté.



Graphe orienté.

Exercice 1

Construire la matrice d'adjacence associée au graphe orienté ci-dessus.
Vous pouvez utiliser `numpy.zeros((`m`,`n`))` pour instancier un tableau (rempli de 0) à deux dimensions avec $m \times n = 0$.

Exercice 2

Écrire une fonction `is_directed(A)` qui renvoie `True` si le graphe associé à la matrice d'adjacence A est orienté. Sinon, il renvoie `False`. Pour cet exercice, n'utilisez aucune méthode de `numpy` !

Exercice 3

Écrire une fonction `remove_edge(`x`, `y`, `A`)` qui supprime l'arête allant du noeud x vers le noeud y dans une matrice d'adjacence A.

Exercice 4

Écrire une fonction `has_path(`x`, `y`, `k`, `A`)` qui renvoie `True` si dans un graphe associé à la matrice d'adjacence A, il existe (au moins) un chemin de longueur k entre les noeuds x et y. Sinon, il renvoie `False`.

Exercice 5

Écrire une fonction `vertex_degree(`x`, `G`)` qui renvoie sous la forme d'un `tuple`, le degré entrant et le degré sortant du noeud x dans le cas d'un graphe orienté, et sous la forme d'un entier le degré du noeud dans le cas d'un graphe non orienté. Il en convient de vérifier si le graphe en paramètre est orienté ou non.

Exercice 6

Un graphe est dit complet si et seulement si tous les sommets sont adjacents deux à deux. Écrire une fonction `is_complete(M)` qui renvoie `True` si la matrice d'adjacence en entrée est associée à un graphe complet. Sinon, il renvoie `False`. N'oubliez pas de prendre en compte le sens des arêtes du graphe.